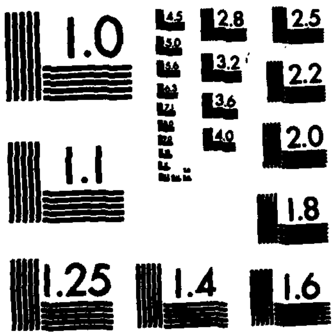
AD-A124 518 HEDGSIM ROUTINES FOR LEADTIME VARIABILITY INVENTORY 1/1
POLICY RESEARCH(U) DECISION SYSTEMS BEAVERCREEK OH
W S DEMMY SEP 81 WP-81-02 F33600-80-C-0530

UNCLASSIFIED F/G 9/2 NL

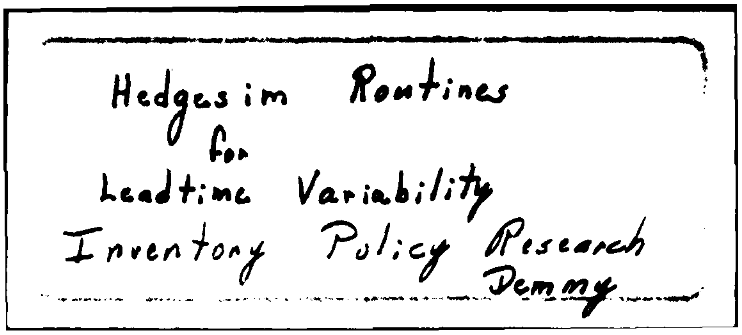
END

FILMED

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



Hedgesim Routines
for
Leadtime Variability
Inventory Policy Research
Demmy

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

DOD LOGISTICS STUDY SUMMARY	1. LD NO.	2. DATE OF SUMMARY

3. TITLE AND ACRONYM NEDGESIM ROUTINES FOR LEADTIME VARIABILITY INVENTORY POLICY RESEARCH

4. STATUS COMPLETED	5. STARTING DATE 06/80	6. COMPLETION DATE 09/81	7. ORGANIZATION REPORT NO. WP-81-02

8. DLSIE SEARCH NO.	9. CONTRACT NO. F33600-80-C-0530	10. TYPE STUDY CONTRACT	11. TIME FRAME	12. COST 35,000

13. SECURITY CLASS UNCLASS	14. DISTRIBUTION LIMITATION UNLIMITED	15. MAN-YEAR EFFORT 1 YR

16. STUDY SPONSOR

RESPONSIBLE INDIVIDUAL

AUTOVON

COMMERCIAL NO.

17. PERFORMING ORGANIZATION

DECISION SYSTEMS
2125 CRYSTAL MARIE DR.
BEAVERCREEK, OH 45431

RESPONSIBLE INDIVIDUAL W. STEVEN DEMMY

AUTOVON

COMMERCIAL NO. 1-513-426-8515

18. ABSTRACT (UNCLASSIFIED)

THIS PAPER DOCUMENTS FORTRAN SOURCE CODE FOR SIMULATING GAMMA LEADTIMES AND NEGATIVE BINOMIAL REQUISITION SIZES IN THE NEDGSIM LONG SUPPLY SIMULATION MODEL.

19. CONCLUSIONS (UNCLASSIFIED)

DTIC
ELECTE
FEB 16 1983
S D
H

20. RECOMMENDATIONS (UNCLASSIFIED)

21. IMPLEMENTING ACTIONS (UNCLASSIFIED)

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

HEDGSIM ROUTINES

for

Leadtime Variability

Inventory Policy Research

by

W. Steven Demmy

September 1981

WP-81-02
Decision Systems
2125 Crystal Marie Drive
Beavercreek, Ohio 45431
(513) 426-8515

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AD	2. GOVT ACCESSION NO. A124518	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) HEDGSIM Routines for Leadtime Variability Inventory Policy Research		5. TYPE OF REPORT & PERIOD COVERED INTERIM
		6. PERFORMING ORG. REPORT NUMBER WP-81-02
7. AUTHOR(s) W. Steven Demmy		8. CONTRACT OR GRANT NUMBER(s) F33600-80-C-0530
9. PERFORMING ORGANIZATION NAME AND ADDRESS Decision Systems 2125 Crystal Marie Drive Beavercreek, Ohio 45431		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Directorate of Management Science Hq.AFLC/XRS Wright-Patterson Air Force Base, Ohio		12. REPORT DATE September 1981
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report)
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This paper documents Fortran Source Code for simulating Gamma Leadtimes and Negative Binomial requisition sizes in the HEDGSIM Long Supply Simulation Model.

Table of Contents

DTIC
COPY
INSPECTED
2

Accession For
NTIS GRA&I ☑
DTIC TAB □
Unannounced □
Justification
By
Distribution/
Availability Codes
Dist | Avail and/or Special
A

Introduction

This paper documents a series of subroutines developed in support of inventory policy research concerning the impacts of lead time variability upon D062 Economic Order Quantity Items. The subroutines are designed to operate within the framework of the HEDGSIM Long Supply Simulation Model. Appendix A contains Job Control Language (JCL) statements required to utilize these new routines with other members of the HEDGSIM simulation program library. On the other hand, Appendix B presents the Fortran source program listings of new routines developed for this study.

The new routines documented in this paper may be *assigned* to one of three different categories. These are: (a) routines for proposed forecasting or safety level calculations, (b) routines for simulating negative binomial requisition sizes, and (c) modified HEDGSIM routines required to implement the above programs.

Let us now consider each of these categories.

Routines for Proposed Forecasting and Safety Level Calculations

This category includes subroutines FOR576, LEVELN, LPGFOR, and SNBROP. Subroutine FOR576 is the same as the original HEDGSIM routine with the addition of an outliers test when the forecasting code ICDFOR = 3. When ICDFOR = 3, the subroutine checks if the largest observed demand exceeds the average demand plus four times the mean absolute deviation (computed using the remaining seven observations). If so, FOR576 assumes the large demand is an "outlier" that is not expected to occur again in future quarters. In this case, the forecasting calculations exclude the outlier data. Subroutines LPGFOR and SNBROP compute optimum reorder levels using the Logarithmic-Poisson-Gamma (LPG) and scaled negative binomial models, respectively. Subroutine LEVELN calls these subroutines and also provides other safety level computation options within the LEVELN logic. As noted above, listings of these programs may be found in Appendix B.

Routines for Negative Binomial Requisition Size Simulation

Subroutines in this category include DEMPAR, NEGBIN, NEGBN1, and RANDEM. These routines implement the generation of negative binomial requisition sizes using techniques described in Reference 2. Subroutine NEGBIN utilizes the twelve most recent observations of actual D062 demand histories to estimate the parameters of a negative binomal distribution of requisition size. It also constructs the cumulative distribution function (CDF) of this distribution. In turn, subroutine NEGBN1 utilizes this CDF to determine the specific requisition size associated with a randomly chosen percentile of the requisition size distribution. Subroutine RANDEM provides a uniform (0,1) ramdom number stream which is used exclusively in the demand generation process. This guarantees that exactly the same sequence of requisitions is generated for a given random number seed regardless of the management policy being evaluated. Finally, subroutine DEMPAR calls subroutines NEGBM1 and RANDEM as needed to generate the specific requisition streams associated with a given demand history. Subroutine NEGBIN is called to initialize the negative binomal requisition size distribution by subroutine INITEM, while the ramdom number generator RANDEM is initialized by the MAIN program at the beginning of a simulation run. Subroutine DEMPAR also includes new logic to insure that very high activity items do not cause the future events list to overflow. With the new logic, after 450 requisitions have been generated, all remaining units of demand associated with a given quarter are placed on one (last) requisition. Only extremely high activity items are impacted by this change.

Modified HEDGSIM Routines

Subroutines included in this category are INITEM, INORD, INVRSM, and ENTERB. As noted above, these routines contain very slight modifications to the original HEDGSIM programs. Subroutine INITEM now contains a call to subroutine NEGBIN to initialize the negative binomal requisition size simulation procedure. Subroutine INORD has been modified to generate GAMMA distributed lead times with a coefficient variation equal to .353. Data collected by Hayya (1980) indicates that such a distribution describes a number of D062 items (See Tables I-1 and I-2). The program INVRSM is essentially identical to the HEDGSIM MAIN program, excepted it now contains a call to initialize subroutine RANDEM.

When simulating high activity items with average requisition sizes that are close to one, very large numbers of requisitions may be generated within the simulation model. If policies are being simulated which offer very low levels of support, this can result in very large numbers of back orders. In turn this may cause the HEDGSIM backorder file to overflow. Subroutine ENTERB prevents this from happening by cancelling all new backorders once the backorder file is filled. With the present coding, once five hundred requisitions have been backordered, all subsequent requisitions are cancelled.

Table I-1

Coefficients of Variation of Lead Times
Observed by Hayya (1980)
for 62 High Activity Items

Coefficient of Variation	Number of Items
.1	8
.2	13
.3	15
.4	11
.5	6
.6	2
.7	2
.8	1
.9	3
1.0	1
	N = 62
	Median = .36

Table I-2

Gamma Probabilities for
Mean = 1 and Coefficient of Variation = .353

x	P(x)	P(X≤x)
.100	.000	.000
.200	.001	.001
.300	.007	.007
.400	.022	.030
.500	.048	.077
.600	.077	.154
.700	.101	.255
.800	.116	.371
.900	.119	.490
1.000	.112	.602
1.100	.098	.700
1.200	.081	.781
1.300	.054	.844
1.400	.048	.892
1.500	.035	.927
1.600	.025	.952
1.700	.017	.969
1.800	.011	.980
1.900	.007	.987
2.000	.005	.992
2.100	.003	.995
2.200	.002	.997
2.300	.001	.998
2.400	.001	.999
2.500	.000	.999
2.600	.000	1.000
2.700	.000	1.000

Note: Underlined values have been incorporated into Subroutine INORD.

References

1. Demmy, W. Steven, HEDGSIM: The Long Supply Simulation Model: Volume II, Program Listing and Narratives, Working Paper 80-10, Decision Systems, 2125 Crystal Marie Drive, Beavercreek, Oh 45431, Dec 1980, 97 pp.

2. Demmy, W. Steven, Modeling the Probability Distribution for Depot-Level Requisition Sizes, Working Paper 80-07, Decision Systems, 2125 Crystal Marie Drive, Beavercreek, Oh 45431, Oct 1980, 160 pp.

3. Hayya, Jack C., Lead Time Variability in Inventory Requirements Projections, Air Force Contract 33615-79-C-5143, Item 0004, Phase 3, Technical Report and Summary, 1962 Norwood Lane, State College, Pa, 16801, June 30, 1980, 71 pp.

APPENDIX A

JOB CONTROL LANGUAGE PROGRAMS

INV.OC.H

INVR.A

```
20       $      IDENT   UP1640,XRS-DENNY     INV.OC.H
30       $      LIMITS  70,,,10K
40       $      NOTE    ***************INSM.JCL DATA FILE FOLLOWS THIS LINE
50       $      OPTION  FORTRAN,NOMAP
60       $      SELECT  INVR/OBJ/INVRSM.O
65       $      SELECT  INVR/OBJ/SNBROP.O
70       $      SELECT  HEDG/OBJ/SIMULA.O
80       $      SELECT  HEDG/OBJ/REQ.O
90       $      SELECT  HEDG/OBJ/CUM.O
100      $      SELECT  HEDG/OBJ/CUMB.O
110      $      SELECT  HEDG/OBJ/DEMPR3.O
120      $      SELECT  HEDG/OBJ/ENTRB3.O
130      $      SELECT  HEDG/OBJ/FILLBO.O
140      $      SELECT  HEDG/OBJ/FOR573.O
150      $      SELECT  HEDG/OBJ/INITAL.O
160      $      SELECT  HEDG/OBJ/INITM3.O
170      $      SELECT  HEDG/OBJ/INORD3.O
180      $      SELECT  HEDG/OBJ/LEVLN3.O
190      $      SELECT  HEDG/OBJ/LONGSP.O
200      $      SELECT  HEDG/OBJ/ORDER.O
210      $      SELECT  HEDG/OBJ/OUTREP.O
220      $      SELECT  HEDG/OBJ/PFAC.O
230      $      SELECT  HEDG/OBJ/PLOTR.O
240      $      SELECT  HEDG/OBJ/RECEIV.O
250      $      SELECT  HEDG/OBJ/RET.O
260      $      SELECT  HEDG/OBJ/STATN2.O
270      $      SELECT  HEDG/OBJ/SSTAT2.O
280      $      SELECT  HEDG/OBJ/ZERO2.O
290      $      NOTE    **********************PRED ROUTINES FOLLOW----------
300      $      NOTE    ****************REQS ROUTINES FOLLOW THIS LINE
310      $      SELECT  REQS/STATUS.O
320      $      SELECT  REQS/FORUPD.O
330      $      SELECT  REQS/LEVEL.O
340      $      SELECT  REQS/ENTER.O
350      $      SELECT  REQS/REMOVE.O
360      $      SELECT  REQS/URIFEL.O
370      $      SELECT  REQS/INFEL.O
380      $      SELECT  REQS/RANDU.O
390      $      SELECT  REQS/GP.O
400      $      EXECUTE
410      $      LIMITS  70,40K,,10K
420      $      PRMFL   07/X1S,R,S,INVR/D062OC.H
430      $      FILE    08,X2S
440      $      FILE    09,A3S
450      $      NOTE    *****SELECTA HEDG/PFAC.D FOR PROGRAM FACTORS
460      $      DATA    04
470      $      SELECTA HEDG/PFAC.D
```

INV.OC.H

```
480     $       DATA    05
490       8060
500      0 1 0 1
510       0 0 0 0 0   0 0
520       0 0 0 0 0   0 0 0 0 0   0 0.0 0 0    0 0 0 0 0
530       7 2 38
540        2 1 2   8 1   2 1
550       6 36   0  99
560        .20  300.  500   19500   .5   1   C7
570       1  30  1 500  8  0    SIMULATE 500 ITEMS FOR 30 QTRS
580       2000
590     $       NOTE    ***************************RIME/GO/PUNCH.E1
600     $       CONVER  NSPIN
610     $       INPUT   NMEDIA
620     $       OUTPUT  MODBCD
630     $       FILE    IN,A3S
640     $       PUNCH   OT
650     $       ENDJOB
```

INV.OC.H

```
20      $     IDENT   WP1596,XRS-DEMMY     INVR.A
30      $     LIMITS  15,,,10K
40      $     NOTE    ***************INSM.JCL DATA FILE FOLLOWS THIS LINE
50      $     OPTION  FORTRAN,NOMAP
60      $     SELECT  INVR/OBJ/INVRSM.O
65      $     SELECT  INVR/OBJ/SNBROP.O
70      $     SELECT  HEDG/OBJ/SIMULA.O
80      $     SELECT  HEDG/OBJ/REQ.O
90      $     SELECT  HEDG/OBJ/CUM.O
100     $     SELECT  HEDG/OBJ/CUMB.O
110     $     SELECT  HEDG/OBJ/DEMPR3.O
120     $     SELECT  HEDG/OBJ/ENTRB3.O
130     $     SELECT  HEDG/OBJ/FILLBO.O
140     $     SELECT  HEDG/OBJ/FOR573.O
150     $     SELECT  HEDG/OBJ/INITAL.O
160     $     SELECT  HEDG/OBJ/INITM3.O
170     $     SELECT  HEDG/OBJ/INORD3.O
180     $     SELECT  HEDG/OBJ/LEVLN3.O
190     $     SELECT  HEDG/OBJ/LONGSP.O
200     $     SELECT  HEDG/OBJ/ORDER.O
210     $     SELECT  HEDG/OBJ/OUTREP.O
220     $     SELECT  HEDG/OBJ/PFAC.O
230     $     SELECT  HEDG/OBJ/PLOTR.O
240     $     SELECT  HEDG/OBJ/RECEIV.O
250     $     SELECT  HEDG/OBJ/RET.O
260     $     SELECT  HEDG/OBJ/STATM2.O
270     $     SELECT  HEDG/OBJ/SSTAT2.O
280     $     SELECT  HEDG/OBJ/ZERO2.O
290     $     NOTE    *********************PRED ROUTINES FOLLOW----------
300     $     NOTE    ****************REQS ROUTINES FOLLOW THIS LINE
310     $     SELECT  REQS/STATUS.O
320     $     SELECT  REQS/FORUPD.O
330     $     SELECT  REQS/LEVEL.O
340     $     SELECT  REQS/ENTER.O
350     $     SELECT  REQS/REMOVE.O
360     $     SELECT  REQS/WRIFEL.O
370     $     SELECT  REQS/INFEL.O
380     $     SELECT  REQS/RANDU.O
390     $     SELECT  REQS/GP.O
400     $     EXECUTE
410     $     LIMITS  15,40K,,10K
420     $     PRMFL   07/X1S,R,S,INVR/D062SM.H
430     $     FILE    08,X2S
440     $     FILE    09,A3S
450     $     NOTE    *****SELECTA HEDG/PFAC.D FOR PROGRAM FACTORS
460     $     DATA    04
470     $     SELECTA HEDG/PFAC.D1
```

INVR.A

```
480     $       DATA     05
490       9030
500      0 1 0 0
510      0 0 1  0  1          0 0
520      0 0 0 0 0    0 0 0 0 1   0 0 0 0 0   0 0 0 0 0
530      7 2 38
540       2 1 2  9 1   2 1
550      6 36       -99   99
560       .20  300.  500   19500   .5   1   C7
570      1 2  1  30  8  0    RUN 2 QTRS, 30 ITEMS
580      25
650     $       ENDJOB
```

*

FORECASTING AND LEVELS CALCULATIONS ROUTINES

FOR576

LEVELN

LPGFOR

SNBROP

CATALOG/FILE DESCRIPTION= HFDG/FOR573.S

```
*#RUN=$HFDG/OBJ/FOR573.O(BCD,NOGO)
*FOR573.S
        SUBROUTINE FOR576(N)
C**************************************************************************
          PARAMETER NQQQ=38
C**************************************************************************
         COMMON/IFBUG/IFBUG
         COMMON/IISMGC/IISMGC
         COMMON/ITIME/ITIME
          COMMON/ITQTR/ITQTR
          COMMON/NDEMAC/NDEMAC(1)
          COMMON/NRETAC/NRETAC(1)
          COMMON/NREQAC/NREQAC(1)
         COMMON/NDEMND/NDEMND(1,NQQQ)
          COMMON/REQSIZ/REQSIZ(1)
         COMMON/NRETUR/NRETUR(1,NQQQ)
          COMMON/NREQ/NREQ(1,NQQQ)
         COMMON/ADR/ADR(1)
         COMMON/RSIGLT/RSIGLT(1)
         COMMON/NDENT/NDENT(1)
         COMMON/LTADM/LTADM(1)
         COMMON/LTPROD/LTPROD(1)
         COMMON/UCOST/UCOST(1)
         COMMON/IDBUG/IDBUG
         COMMON/ICDFOR/ICDFOR
         COMMON/ICDSIG/ICDSIG
          LW=0
         IDUM=0
          IFUM=0
         IRUM=0
          KK=NDENT(N)
          IF(KK.GT.8)KK=8
          KL=KK-1
          DO 1  I=1,KL
         IDUM=IDUM + NDEMND(N,I)
          IFUM=IFUM + NREQ(N,I)
       1 IRUM=IRUM + NRETUR(N,I)
C
          IT=MOD(ITIME,ITQTR)
          R=FLOAT(IT)/FLOAT(ITQTR)
          RM=1.-R
          DUM=R*FLOAT(NDEMAC(N))+FLOAT(IDUM)+RM*FLOAT(NDEMND(N,KK))
          FUM=R*FLOAT(NREQAC(N))+FLOAT(IFUM)+RM*FLOAT(NREQ(N,KK))
          RUM=R*FLOAT(NRETAC(N))+FLOAT(IRUM)+RM*FLOAT(NRETUR(N,KK))
C
C
*
**:COMPUTE LEVELS BASED ON NET DEMANDS, WHERE NET IS GROSS
*::         DEMAND MINUS SERVICEABLE RETURNS FOR THE PAST 24
*::         MONTHS,
*::RETURNS ARE ADDITIONS TO ON-HAND ASSETS.
```

FOR573.S
FOR576

```
520*
530      FOR=0.
540      IF(IISMGC.GE.3)FOR = FOR-R.
550        IF(FOR.LE.0.)GO TO 1000
560       FORCST=FOR/FLOAT(KK)
570 900           AFORCD=FORCST*4.
580C
590C**COMPUTE MAD OF QUARTERLY DEMAND
600C
610        QMAD=0.
620C
630C             IF IISMGC =>3,
640C        FORCST AND QMAD ARE BASED ON NET QTRLY DEMAND RATE
650C             OTHERWISE, BOTH ARE BASED ON GROSS DEMANDS
660C
670      KZERO=0
680      DMAX=-999999.
690        DO 20 I=1,KK
700        DEM=NDEMAD(N,I)
710        IF(IISMGC.GE.3)DEM=DEM-NRETUR(N,I)
720      IF(DEM.GT.0)KZERO=KZERO+1
730      IF(DEM.LE.DMAX)GO TO 10
740         DMAX=DEM
750      IDMAX=I
760   10 CONTINUE
770        ABSDEV=ABS(DEM-FORCST)
780        QMAD=QMAD+ABSDEV
790   20 CONTINUE
800C
810      DEM=NDEMAC(1)
820      IF(IISMGC.GE.3)DEM=DEM-NRETJR(N,I)
830      CMAD=ABS(DEM-FORCST*R)*R
840        QMAD=(QMAD+CMAD)/(FLOAT(KK)+R)
850        IF(QMAD.LT. 0.01) QMAD=0.01
860C
870C             IF FORECAST CODE=3, BACK OUT THE LARGEST
880C                 DEMAND THAT EXCEEDS 4 MADS.
890C
900      IF(ICDFOR.NE.3)GO TO 40
910C
920C             CHECK IF FILTER IS TO BE USED
930C
940      IF(KZERO.LE.3)GO TO 40
950      IF(QMAD.LT.1.)GO TO 40
960C
970C      IF DMAX>TFOR +1*MAD,
980C             THEN BACK OUT BIGGEST DEMAND.
990C
1000C
1010      TFOR=(8.*FORCST-DMAX)/7.
1020C
```

FOR573.S
FOR576

```
1030C          COMPUTE 7-PT MAD
1040C
1050      TMAD=0.
1060      DO 30 I=1,KK
1070       IF(I.EQ.IDMAX) GO TO 30
1080       DEM=NDEMND(N,I)
1090       IF(IISMGC.GE. 3) DEM=DEM-NRETUR(N,I)
1100      TMAD=TMAD + ABS( DEM -TFOR)
1110   30 CONTINUE
1120      TMAD = TMAD/7.
1130C
1140C
1150C          IS DMAX AN OUTLIER?
1160C
1170      IF(DMAX.LE.(TFOR+4.*TMAD))GO TO 40
1180C
1190C          USE BACKED-OUT VALUES
1200C
1210      FORCST=TFOR
1220      ADR(N) =4.*FORCST
1230      QMAD=TMAD
1240      IF(IFBUG.GE.1)WRITE(6,33)DMAX
1250   33 FORMAT('****FOR576--BACK OUT OUTLIER OF ',F8.1,' UNITS')
1260C
1270   40  CONTINUE
1280C
1290C**COMPUTE STANDARD DEVIATION OF LEADTIME DEMAND
1300C
1310        RLT=FLOAT(LTADM(N)+LTPROD(N))
1320        RSIGLT(N)=0.5945*QMAD*(0.82375+0.42625*RLT)
1330        IF(RSIGLT(N).LT. 0.01) RSIGLT(N)= 0.01
1340C
1350C**COMPUTE AVERAGE REQUISITION SIZE
1360C
1370        IF(FUM.LT.1)FUM=1
1380        R=DUM/FUM
1390        IF(R.LT.1.)R=1.
1400        REQSIZ(N)=R
1410       GO TO 2000
1420C
```

```
C            TREATMENT FOR ZERO-DEMAND ITEMS
C
 1000 RSIGLT(N)=.5
      ADR(N)=0.0
     REQSIZ(N)=1.
      GO TO 2000
 2000 IF(IFBUG.LT. 1) RETURN
      WRITE(LW,100)
      WRITE(LW,200) N,NDENT(N),ICDFOR,ICDSIG,LTADM(N),LTPROD(N)
      WRITE(LW, 300)
       WRITE(LW,200)NDEMAC(N),(NDEMND(N,J),J=1,KK)
       WRITE(LW,200)NRETAC(N),(NRETUR(N,J),J=1,KK)
       WRITE(LW,200)NREQAC(N),(NREQ(N,J),J=1, KK)
      WRITE(LW,400)
      WRITE(LW,500) FORCST,ADR(N),REQSIZ(N),UCOST(N),QMAD,RSIGLT(N)
  100       FORMAT(1H ,4X,'****FOR573--N,NDENT,ICDFOR,ICDSIG',
&         '  LTADM,LTPROD')
  200 FORMAT(1H ,4X,12I7/12I7)
  300       FORMAT(1H ,9X,'DEMAND, RETURNS, AND FREQ',
&         ' IN CURRENT HISTORY FILE')
  400       FORMAT(1H ,'     FORCST          ADR    AVE-REQ     UCOST',
&         '      QMAD     RSIGLT')
  500       FORMAT(1H ,6F10.4)
      RETURN
      END
```

FOR573.S
FOR576

```
CATALOG/FILE DESCRIPTION=  HEDG/LEVLN3.S

10*#RUN=$HEDG/OBJ/LEVLN3.O(SCL,NOGO)
20*LEVELN3.S
30          SUBROUTINE LEVELN(N)
40C
50C            THIS ROUTINE COMPUTES        REORDER,STOCK OBJECTIVE,RETENTION,
60C               TERMINATION, AND SUPPORT LEVELS.
70C
80         PARAMETER NQQQ=33
90C
100C
110          COMMON/IHBUG/IHBUG
120         COMMON/ICDFOR/ICDFOR
130          COMMON/COSHRT/COSHRT
140          COMMON/COSHLD/COSHLD
150          COMMON/COSORD/COSORD(3)
160          COMMON/CSTBRK/CSTBRK
170          COMMON/GSULF/GSULF
180          COMMON/GRLF/GRLF
190          COMMON/GSLF/GSLF
200          COMMON/GTLF/GTLF
210           COMMON/ITINV/ITINV
220          COMMON/ITLEVL/ITLEVL
230          COMMON/IDLEVL/IDLEVL
240          COMMON/NITEM/NITEM
250          COMMON/NDHIS/NDHIS
260          COMMON/POLICY/ICDEOQ,ICDSL,EOQMAX,EOQMIN,SLMAX,SLMIN,RLF,TLF,SU
270           COMMON/ICDSLL/ICDSLL
280          COMMON/ADR/ADR(1)
290          COMMON/LTADM/LTADM(1)
300          COMMON/LTPROD/LTPROD(1)
310          COMMON/IROTY/IROTY(1)
320          COMMON/IRL/IRL(1)
330          COMMON/IROL/IROL(1)
340          COMMON/ISUL/ISUL(1)
350         COMMON/ISLEVL/ISLEVL(1)
360          COMMON/ITL/ITL(1)
370          COMMON/RSIGLT/RSIGLT(1)
380          COMMON/REQSIZ/REQSIZ(1)
390          COMMON/UCOST/UCOST(1)
400           COMMON/PF/PF(NQQQ),PFA(NQQQ,30),ACODE(30),NCODE,AIRCRF
410           COMMON/IISMGC/IISMGC
420          DATA Z/1.0/
430          DATA ITLMIN,ITLMIN/99999999,99999999/
440C
450C           COMPUTE PLANNING FACTORS
460C
470          AMDA=COSHRT
480          RLTM=LTADM(N)+LTPROD(N)
490          IF(RLTM.LT.1.)RLTM=0.5
```

```
C
       AD=ADR(N)
      SIG = RSIGLT(N)
C
C             DETERMINE PROGRAM FACTOR PFNOW
C
       IQTR=ITINV
      PFNOW=1.00
C
C                   IF ICDFOR .EQ. 2, ADJUST FOR FORECAST PROGRAM CHANGES
C
       IF(ICDFOR.NE.2) GO TO 20
C
      PFNOW = PF(IQTR + NOHIS )
C
       AD = PFNOW * AD
      SIG=SIG * (PFNOW ** 0.85)
C
C
   20 CONTINUE
C
       RLT=AD*RLTM/12.
       RMR=AD/12.
       UC=UCOST(N)
       ADDR=UC*AD
C
C      SET COR EQUAL TO SMALL PURCHASE ORDER COST
       COR=COSORD(1)
C
C        BRANCH BY EOQ FORMULA     CODE
C
       GO TO (110,120,130,140,150,160,170,180,190),ICDFOQ
C
C        2 FEB      70 GUIDELINE LTR POLICY
C
  110       IF(ADDR.GE.1000.) GO TO 62
       Q=12.*RMR
       GO TO 200
   62 IF(ADDR.GE.5000.) GO TO 64
       Q=6.*RMR
       GO TO 200
   64 Q=3.*RMR
       GO TO 200
C
C             AFLC 57-6 POLICY---DEC 1976
C
  120       Q=SQRT(2.*COR*AD/(COSHLD*UC))
       ACOST=Q*UC
       IF(ACOST.LT.CSTBRK) GO TO 200
       COR=COSORD(2)
        Q=SQRT(2.*COR*AD/(COSHLD*UC))
        GO TO 200
```

```
 130       CONTINUE
 140       CONTINUE
 150       CONTINUE
 160       CONTINUE
 170       CONTINUE
 180       CONTINUE
 190       CONTINUE
      WRITE(6,8195)ICDEOQ,ICDSL
 8195  FORMAT('****LEVELN--UNDEFINED FORMULA CODE....',
&     ' ICDEOQ=',I3,'  ICDSL=',I3)
      STOP
C
C       CHECK     ORDER SIZE LIMITS
 200  CONTINUE
      EMX=EOQMAX*RMR
      IF(Q.GT.EMX) Q=EMX
      EMX=EOQMIN*RMR
      IF(Q.LT.EMX) Q=EMX
      IF(Q.LT.1.)Q= 1.
C
C       BRANCH BY SAFETY LEVEL FORMULA CODE ICDSL
C
      GO TO (310,320,330,340,350,360,370,380,390),ICDSL
C
C          USE 12 MONTH SUPPLY
C
 310   CONTINUE
        Q= 12.*RMR
C
      GO TO 500
C
C       23 AUG 68 GUIDELINE LTR POLICY
C
 320  SL=0.25*RLT
      GO TO 500
C
C       AFLCM     57-6 POLICY
C
 330       Z=SQRT(REQSIZ(N))
      GO TO 358
C
C          PT-FORMULA TO MINIMIZE UNITS BACKORDERED
C
 340       Z=1.
      GO TO 358
C
```

```
C             PT-FORMULA TO MINIMIZE REQUISITIONS BACKORDERED
C
  350         Z=REQSIZ(N)
C
C
  358   CONTINUE
        IF(SIG.LE.0.)SIG=.0001
        QSIG=-1.414*Q/SIG
        RNUM=AMDA*SIG
        IF((QSIG.LT.0.).AND.(QSIG.GT.-30.))
&             RNUM=AMDA*SIG*(1.-EXP(QSIG))
         DNOM=2.*COSHLD*UC*Z*1.414*Q
         RK=0.707*ALOG(RNUM/DNOM)
         SL=RK*SIG
         GO TO 500
C
C
C             MODIFIED STD.DEV. FOR ACTUAL GAMMA LEAD TIME VARIABILITY
C
C             ASSUME COFF OF VARIATION =.353. AND I PERIOD =EXPECTED LEA
C
  360  CONTINUE
         RMEANL=1.
         SIGL=.353*RMEANL
         VARL=SIGL**2
         VARD=SIG**2
         SIGOLD=SIG
C
C             REVISE SIG FOR LT VARIABILITY
C
        SIG=SQRT(RMEANL*VARD+(RLT**2)*VARL)
        IF(IHBUG.GT.0)WRITE(6,363)SIGOLD,SIG,SIG/SIGOLD
C
  363  FORMAT(T20,"SIGOLD=",F10.1,"   SIG LTD=",F10.1,"   RATIO=",F10
C
C             NOW USE D062 SQRT(Z)FORMULA
C
        GO TO 330
C
***************************************************
C
C           SCALED NEGATIVE BINOMIAL MODEL
C
  370   CONTINUE
C
```

LEVLN3.S
LEVELN

```
       IROP = 0
       IF(AD.LE.0.) GO TO 377
       IF(AMDA.LE.0.) GO TO 377
C
C            COMPUTE CRITICAL STOCK STOCK OUT PROBABILITY
C
      POUT=(Q*COSHLD*UC)/(AMDA*AD)
C
C            FIND CRITICAL FILL PROBABILITY
C
      FCRIT=1.-POUT
      IF(FCRIT.LE.0.) GO TO 377
C
C            AVERAGE REQUISITION SIZE
C
           AVEREQ=REQSIZ(N)
C
C             CUSTOMERS PER PERIOD
C
      ED=RMR
C
C            E(LT)
C
           ELT=FLOAT(LTADM(N) + LTPROD(N) )
C
C            ASSUME COEFF. OF VAR. OF LT=.353
C
      CVLT=.353
C
C            FIND SCALED NEG. BIN. REORDER POINT
C
      CALL SNBROP(AVEREQ,ED,ELT,CVLT,FCRIT,IROP)
C
C            BACK INTO THE SAFETY LEVEL
C
  377  CONTINUE
      SL=IROP-RLT
C
C
C            SKIP SL LIMITS
C
      GO TO 600
*FILE EXPFIT.S
C-----------------------------------------
C                    EXPONENTIAL APPROXIMATION MODEL
C
  380 CONTINUE
      IF(AD.LE.0.) GO TO 387
      IF(AMDA.LE.0.) GO TO 387
C
```

```
C              COMPUTE OPTIMAL STOCKOUT PROBABILITY
C
        POUT = (Q*COSHLD*UC)/(AMDA*AD)
C
C              LIMIT OPTIMUM FILL PROBABILITY
C
        PFILL = 1.-POUT
C
        IF(PFILL.LE.0.) GO TO 387
        IF(PFILL.GT. 0.9999) PFILL = 0.9999
C
C              ASSUME P(Z <= ZT) = F(Z) = 1.-.331 *EXP( B*ZT)  FOR ZT >
C
        ZT = 0.
        IF(PFILL .GT. 0.669) ZT = ALOG( (1.-PFILL)/.331 )/(-.463)
C
C              COMPUTE MAD, AND SAFETY LEVEL FROM MAD
C
C              SINCE SIGLT = .5945*QMAD*(.8235+0.42625*RLTM), WE HAVE
C
        QMAD = SIG/(.5945*(.82375+0.42625*RLTM) )
C
C              NOW COMPUTE SAFETY LEVEL
C
        SL = ZT * QMAD * SQRT (RLTM)
C
C
C              SKIP THE LIMIT CALCULATION
C
        GO TO 600
C
C              SET SL TO ZERO
C
     387 CONTINUE
        SL = 0.
        GO TO 600
C
C-----------------------------------------------------------
     390       CONTINUE
C
C              USE SPLIT-EXPONENTIAL-GAMMA MODEL
C-----------------------------------------------------------
        IF(AD.LE.0.) GO TO 397
        IF(AMDA.LE.0.) GO TO 397
C
C              COMPUTE OPTIMAL STOCKOUT PROBABILITY
C
        POUT = (Q*COSHLD*UC)/(AMDA*AD)
C
```

LEVLN3.S
LEVELN

```
C              LIMIT OPTIMUM FILL PROBABILITY
C
       PFILL = 1.-POUT
C
       IF(PFILL.LE.0.) GO TO 397
       IF(PFILL.GT. 0.9999) PFILL = 0.9999
C
C
C              ESTIMATE QUARTERLY DEMAND RATE, MAD, AND LEADTIME
C
C              SINCE RSIGLT = .5945*QMAD*(.8235+0.42625*RLTM), WE HAVE
C
       QMAD = SIG/(.5945*(.82375+0.42625*RLTM) )
       QRATE = AD/4.
       QTRLT = RLTM/3.
C
C              NOW COMPUTE REORDER LEVEL
C
       CALL EXPGAM(QRATE,QMAD,QTRLT,PFILL,IROP)
C
       IF(IROP.LT.0) IROP=0
       SL=IROP-RLT
C
C
C              SKIP THE LIMIT CALCULATION
C
       GO TO 600
C
C              SET ROP TO ZERO
C
  397 CONTINUE
       IROP=0
       SL=IROP-RLT
       GO TO 600
C
************************************************************************
*****
  500          CONTINUE
C
C              BRANCH BY SAFETY LEVEL LIMIT CODE, ICDSLL
C
          GO TO (510,520),ICDSLL
C
C              D062 LIMITS, AS OF JUNE 77
C
  510          CONTINUE
C
```

```
C               LIMIT SAFETY LEVEL
C
C
C               LIMITS SL TO LESS THAN LEADTIME DEMAND
C
          SLM=RLT
          IF(SL.GT.SLM)SL=SLM
C
C               LIMIT SL TO LESS THAN 3*SIG
C
          SLM=3.*SIG
          IF(SL.GT.SLM)SL=SLM
**
C
C               LIMITS BASED ON MONTHS OF SUPPLY
C
  520           CONTINUE
C               LOWER LIMIT
           SLM=SLMIN*RMR
           IF(SL.LT.SLM)SL=SLM
C               UPPER LIMIT
          SLM=SLMAX*RMR
           IF(SL.GT.SLM)SL=SLM
C
C
           GO TO 600
C***************************************************************
C
  600           CONTINUE
C
C         COMPUTE LEVELS
C
          ISLEVL(N)=SL+0.5
          IROTY(N)=Q+0.5
          IROL(N)=SL+RLT+0.5
           IF(IROL(N).LT.0) IROL(N) = 0
          ITL(N) =SLMAX*RMR+RLT+GTLF*RMR+0.5
          IF(ITL(N).LT.0) ITL(N)=0
          IRL(N)=FLOAT(ITL(N))+GRLF*RMR+0.5
          IF(ITL(N).LE.ITLMIN) ITL(N)=ITLMIN
          IF(IRL(N).LE.IRLMIN)IRL(N)=IRLMIN
          ISUL(N)=GSULF*RMR+0.5
 3000     CONTINUE
          ROL=FLOAT(IROL(N))
          IF(IHBUG.LT.1) RETURN
         RK=0.
         IF(SIG.GT.0.) RK = SL/SIG
          WRITE(6,8903)N,IROTY(N),IROL(N),ITL(N),IRL(N),ISUL(N),RK,
&             SL,PFNOW,AD,PFILL
 8903      FORMAT(4X,'****LEVELN--N=',I5,'  IROTY=',I5,'  IROL=',I5,
&         '  ITL=',I5,'  IRL=',I5,'  ISUL=',I5,'   RK=',F9.4,
&         /,T50,'   SL=',F9.1,'   PFNOW=',F6.3,'   AD=',F9.1
&           ,'     PFILL =',F6.3 )
C
          RETURN
          END
```

LEVLN3.S
LEVELN

```
      SUBROUTINE EXPLTD(X,R,QMAD,QTRLT,CUMPX)
C
      COMMON/IWT/IWT(20)
C
C           SET WRITE FLAGS
C
      IDETL = IWT(10)
      IPNTSZ = IWT(11)
      CUMPT=0.
      CUMPX=0.
C
C           INITIALIZE PDF PARAMETERS
C
      A1= 0.331
      B1 = -0.463
C
      A2= 0.069
      B2= 0.7979
C
C           GAMMA CONSTANT FOR MEAN=1 AND CV=.353
      C1= 0.0015373
C
C
C------------------------------------------------
C
C
C           INITIALIZE FOR T INTEGRATION
C
      DT = .1
      T = DT
      CUMPT=0.
      CUMPX=0.
C
C           BEGIN "T" INTEGRATION LOOP
C
      DO 100 I=1,100
C
C           COMPUTE STANDARDIZED ERROR ZT
C
      TQTR= T*QTRLT
      ZT = (X - R*TQTR)/ (QMAD*SQRT(TQTR) )
C
```

```
C            COMPUTE P( T)
C
      GT = C1*(8.*T)**7* EXP(-8.*T) * DT
C
C            COMPUTE P(Z <= ZT & T)
C
      IF( ZT.LE.0.) PZ =A2*EXP(B2*ZT)
      IF( ZT.GT.0.) PZ= 1. - A1*EXP( B1*ZT)
C
C            COMPUTE P( X <= X &T) P( T)
C
      PXT = PZ*GT
C
C            UPDATE CUMULATIVE PROBABILITIES
C
      CUMPT = CUMPT + GT
      CUMPX = CUMPX + PXT
C
      IPRNT=0
      IF( (IPNTSZ.GT.0) .AND. (MOD(I,IPNTSZ) .EQ. 0) )
     &       IPRNT=1
      IF(IDETL.LE.0) IPRNT=0
      IF(IPRNT.GT.0) WRITE(6,63) X,T,GT,PXT,CUMPT,CUMPX,ZT,PZ
   63 FORMAT(2F8.2,6F10.4)
C
C
C             INCREMENT T
C
      T = T+DT
C
C             STOP IF CUMPT > .999
C
      IF(CUMPT.GT. 0.999) GO TO 120
C
C--------------END OF "T" LOOP--------
  100 CONTINUE
C
  120 CONTINUE
      RETURN
      END
      SUBROUTINE EXPGAM(QRATE,QMAD,TOTR,PFILL,IROP)
C
C              COMPUTE A RE-ORDER POINT (IROP) TO GIVE
C                A FILL RATE OF PFILL.
C
      COMMON/IWT/IWT(20)
C
C              SET LOW POINT FOR P.D.F
C
      QMADLT = QMAD*SQRT(TOTR)
      XLOW = QRATE*TOTR - 5.*QMADLT
      XLOW = IFIX(XLOW + 0.5)
      IF(XLOW .LT.0.) XLOW = 0.
      CALL EXPLTD(XLOW,QRATE,QMAD,TOTR,CUMPX)
C
      IF(PFILL.GT.CUMPX) GO TO 500
C
C           VERY LOW SUPPORT NEEDED.
C              SET REORDER POINT TO XLOW
C
      IROP = XLOW
      RETURN
C
C              SET LOW POINT FOR SEARCH
C
  500 CONTINUE
      PLOW= CUMPX
C
C              COMPUTE DELTA-X
C
     DX = QMADLT
      IF(DX.LT.1.) DX = 1.
      IF(DX.GT.1.) DX = IFIX( DX + 0.5)
C
C              INITILIZE HIGH VALUES FOR SEARCH
C
     X= XLOW + DX
C
C              FIND P(DLT <= X)
C
  510 CONTINUE
```

```
      CALL EXPLTD(X,DRATE,DMAD,TOTD,CUMPX)
      IF(IWT(10).GT.0) WRITE(6,523)X,CUMPX,XLOW,PLOW
  523 FORMAT("---EXPGAM. X=",F8.1,"   CUMPX=",F8.4,
&        "   XLOW =",F8.1,"   PLOW =",F8.4)
C
C
C             HAVE WE BOUNDED THE DESIRED FILL RATE
C              IF SO, GO TO 600 AND INTERPOLATE FOR ROP.
C
      IF(PFILL.LE. CUMPX ) GO TO 600
C
C            IF CUMPX > .99, STOP ANYWAY.
C
      IF(CUMPX.GE. 0.99) GO TO 580
C
C             NO, WE HAVE TO KEEP TRYING. RESET FOR SEARCH
C
      XLOW = X
      PLOW = CUMPX
C
      X= X + DX
      GO TO 510
C
C             SET IROP = THIS X
C
 580  CONTINUE
      IROP = X+ 0.5
      RETURN
C
C             THE ROP IS BOUNDED.  NOW INTERPOLATE.
C
 600  CONTINUE
C
      X = XLOW + (PFILL - PLOW)* ( X - XLOW)/ (CUMPX - PLOW)
C
      IROP = IFIX( X + 0.5)
C
      RETURN
      END
```

```
CATALOG/FILE DESCRIPTION=   INVR/LPGFOR.S


5*FILE  LPGFOR.S
06C
7C              FORTRAN VERSION OF LPGFOR.S
08C
10C
15      CHARACTER ADOLR*1
16C
20C
22C
25          CHARACTER ADOLR*1
30          COMMON/LPGPRM/T1,RL1,A1,B1,ID7,ID8
40C
50C         T(K+1)  = T(X,K) FOR THE CURRENT X
60C        T2(K+1) = T(X-1,K) FROM PREVIOUS X CALCULATION
70C
80C
90C              READ(5,13)  PARAMETERS
91C
92  90 CONTINUE
92      PRINT , "ID7,ID8 "
93      READ(5,13)  ID8,  ID7
94C
95  13 FORMAT (V)
100     CALL LPG0
110C               DO LPG RECURSION
120     CALL LPGREC
130C           DO SCALED NEG BINOMIAL CALCULATIONS
140     CALL LPG3
142C
143C            CALL NAHMIAS EXACT LPG ROUTINE
144C
145     CALL LPG1
146C
150     WRITE(6,13) "CONTINUE (Y OR N)"
155     READ(5,13)  ADOLR
160     IF( ADOLR.EQ."Y" )GO TO 90
170     STOP
171     END
172C
173C
174     SUBROUTINE LPG0
175         COMMON/LPGPRM/T1,RL1,A1,B1,ID7,ID8
177  13 FORMAT(V)
178C
179     DIMENSION T(200),T2(200)
180C
181     WRITE(6,13)
182    WRITE(6,13)
190 WRITE(6,13) "LPG0.S--EXACT LPG PROB CALCULATIONS USING RECURSION"
191     WRITE(6,13)
192     WRITE(6,13)
```

```
200C
210  WRITE(6,13) "READ(5,13)  AVE REQ, E(D), E(LF), C.OF.V OF LF "
220  READ(5,13)  RO,D1,E1,C
230     WRITE (3,13) RO,D1,E1,C
240C
250C                        SOLVE FOR THETA = T1
260     CALL BSRCH (RO,Q)
270   T1=Q
280C
290C                  ESTIMATE PARAMETERS FOR LPG
300C               S = STD DEV,  B = VAR TO MEAN RATIO
310    S = C*E1
320    B = S*S/E1
330     U1= D1*E1
340C
350   B1=1/B
360   A1 =B1*E1
370   R1=-T1/( (1-T1) *ALOG(1-T1) )
380   RL1 = U1/(E1*R1)
390C
400   WRITE(6,13) "AVE REQ SIZE =", R1
410 WRITE(6,13)
420    WRITE(6,13) "THETA =",T1,"LAMBDA =",RL1
430    WRITE(6,13) "ALPHA=",   A1,"BETA =",B1
440C
450C
460    RM1=-T1/((1-T1)*ALOG(1-T1)*RL1*A1/B1)
470 WRITE(6,13)
480 WRITE(6,13) "MEAN =",RM1
490 WRITE(6,13)
500C      ESTIMATE THE FIRST FOUR MOMENTS RM1,RM2,RM3,RM4 OF THELPG DISTRIB
510     D1 = B1*(1-T1)
520    C4 = -RL1*T1/(ALOG(1.-T1))
530C
540C
550     RM1 = A1*C4/D1
560C
570     RM2 = A1*C4*(B1+C4)/D1**2
580C
590     RM3 = A1*C4*((B1**2)*(1+T1) +2*C4**2)/D1**3
600C
610    RM4 = B1**3*(1+4*T1+T1**2) + B1**2*C4*(3.*A1 + 1.)
620     RM4 = RM4 + 6*B1*C4*C4*A1 +C4**3*(3*A1+6. )
630     RM4 = A1*C4*RM4/D1**4
640C
650     WRITE(6,13) "MOMENTS"
660     WRITE(6,13) " RM1       RM2      RM3      RM4"
670     WRITE(6,13) RM1,RM2,RM3,RM4
680C
690C              COMPUTE STANDARDIZED MOMENTS
700     WRITE(6,13)
710     S = SQRT(RM2)
720     WRITE(6,13) "COEF OF VAR =", S/RM1
730     WRITE(6,13) "RM3/S**3= ", RM3/S**3
740     WRITE(6,13) "RM4/S**4= ", RM4/S**4
750C
760C              OUTPUT LPG PARAMETERS TO FILE
770C
780          WRITE (8,13) RM1,S/RM1,RM3/S**3,RM4/S**4
790          WRITE (8,13) T1,RL1,A1,B1
800          RETURN
801     END
```

LPGFOR.S

```
          SUBROUTINE LPGREC
C     **************************************************************
C                        LPG RECURSION CALCULATION
C       ***********************************************************
C
       COMMON/LPGP99/T1,RL1,A1,B1,I07,I08
      DIMENSION T(200),T2(200)
C
C                        REDEFINE C FOR PROBABILITY CALCULATIONS
C
       WRITE(6,13)" ***LPGREC,T1,RL1,A1,B1 ",T1,RL1,A1,B1
       C= -RL1/ALOG(1.-T1)
       C1 = C/(RL1 + B1)
C
C                        SET LIMIT ON X = 200
C
      RL2=200
C
      C2=0
C
C                        SET CONSTANTS FOR USE IN RECURSION
C
       S7=1
      H1=(B1/(RL1+B1))**A1
    H1=S7*H1
C                           S7 = SCALE FACTOR
       H2=(C/(RL1+B1))
C
    WRITE(6,13)"C,C1,H1,H2",C,C1,H1,H2
      13 FORMAT(V)
C
C                          FVALUATE H(X=0)
C
      T(1) = 0
       T2(1) = H1
       S1 = H1
         C2 = C2 + S1/S7
C
      WRITE(6,13)
         WRITE(8,13) 0,S1,C2
C
         WRITE(6,13)
         WRITE(6,13) "            X                 H(X)               F(X)"
         WRITE(6,13)
       WRITE(6,13) 0,S1,C2
C
C
C                    EVALUATE H(X) FOR X .GT. 0
 1200 CONTINUE
         L2=RL2
       DO 1650 IX=1,L2
         X=IX
         IXP1=IX+1
       S1 = 0
         T(1)=0
```

LPGFOR.S
LPGREC

```
1240C
1250       T(IX+1)=(T1*H2*(A1+X-1)/X)*T2(IX)
1260C
1270      IF( X .LT. 2 )GO TO 1380
1280C
1290C
1300C
1310       DO 1360 K=1,IX-1
1320       T(K+1)=(T1/X) * (H2*(A1+K-1)*T2( K) +(X-1)*T2(K+1))
1330       S1=S1+T(K+1)
1340           IF( ID8 .LE. 0 )GO TO 1360
1350       WRITE(6,13)"X,K,T(X,K),S1 ",X,K,T(K),S1
1360  1360 CONTINUE
1370C                     PICK UP T(X,X) TERM IN SUM
1375 1380 CONTINUE
1380       S1=S1 +T(IX+1).
1390C
1400C                     WRITE(6,13) TOTALS FOR H(X)
1410C
1420        C2 = C2 + S1/S7
1430       WRITE(6,13) IX,S1,C2
1440       WRITE(8,13) IX,S1,C2
1450C
1460C                     IF( CUM PROB EXCEEDS .99, STOP
1470C
1480        IF( C2 .GT. .99 )GO TO 1670
1490C
1500C                     WRITE(6,13) T(K) TERMS FOR DEBUGGING
1510C
1520       IF( ID8 .LE. 0 )GO TO 1560
1530       DO 1550 KK=1,IXP1
1535        K=KK-1
1540       WRITE(6,13) "X,K,T(X,K) =>",X,K,T(KK)
1550 1550 CONTINUE
1560 1560 CONTINUE
1570C
1580C                     RECORD T(K) VALUES FOR USE IN NEXT PASS
1590C
1600       DO 1620 KK=1,IXP1
1610         T2(KK)=T(KK)
1620 1620 CONTINUE
1630C
1640C       ----------------------END OF X LOOP
1650 1650 CONTINUE
1660C
1665 1670 CONTINUE
1670       WRITE (8,13) -99,-99,-99,"    END OF LPG"
1680       RETURN
1683       END
1690C       ****************************************END OF LPG RECURSION
```

LPGFOR.S
LPGREC

```
1700C
1710        SUBROUTINE BSRCH(R0,Q)
1720C
1721              COMMON/ZLPGPR/AZF1,BF1,A1,B1,IF7,ID8
1730C      ---------------------------------------AVE REQ SIZE EQUATION
1740C---------- FUNCTION FNR(Q) = -.Z( (1-Q)*ALOG(1-Q)   )
1744C
1745C             SEE 2182-2190 FOR FNR(Q) DEFINITION.
1748C
1750C
1760C      -----------------------------------
1770C                            BINARY SEARCH ROUTINE
1780C       -----------------------------------
1790C
1800C             SOLVE FOR THE VALUE OF Q WHICH GIVES AND
1810C               AVE. REQUISITION SIZE OF R0
1820C
1830C            SET UP END POINTS FOR SEARCH
1840C
1850      K= 0
1860     Q9=.999
1870     R9=FNR(Q9)
1880     Q1=.001
1890     R1=FNR(Q1)
1900       GOTO 2080
1910C
1920C            CHECK IF( R0 .LE. R
1930C
1935 1940 CONTINUE
1940       IF( R0 .GT. R )GO TO 2000
1950C
1960C                     RE-SET TOP OF INTERVAL
1970       R9=R
1980       Q9=Q
1990        GOTO 2080
1991C
1992 2000 CONTINUE
2000C
2010C                      RE-SET BOTTOM OF INTERVAL
2020       R1=R
2030        Q1=Q
2040         GOTO 2080
2050C
2060C                     HALVE THE INTERVAL, AND RE-EVALUATE THE FUNCTION
2070C
2075 2080 CONTINUE
2080        K=K+1
2090       Q=(Q1+Q9)/2
2100        R=FNR(Q)
2120C
2130C                     IF( R IS WITHIN .001 OF R0, )GO TO RETURN
2140C
2141       IF(ID8.GT.0) WRITE(6,23) K,R0,R,Q
2142   23   FORMAT(" ***BSRCH--K,R0,R,Q",I5,3F8.3)
2150        IF( ABS(R-R0) .LT. .001 )GO TO 2180
2160        IF( K .GT. 15 )GO TO 2180
2170         GOTO 1940
2175 2180 CONTINUE
2180      RETURN
2181       END
```

LPGFOR.S
BSRCH

```
  FUNCTION FNR(Q)
   FNR=-Q/((1-Q)*ALOG(1-Q))
 RETURN
 END
```

LPGFOR.S
BSRCH

```
      SUBROUTINE LPG1
DIMENSION Y(100,100)
      COMMON/LPGPRM/T1,RL1,A1,B1,ID7,ID8
WRITE(6,13) "THIS PROGRAM COMPUTES EXACT PROB FOR THE LPG DIST"
 13 FORMAT(V)
WRITE(6,13) "THETA =",T1,"LAMBDA =",RL1,"ALPHA =",A1,"BETA =",B1
      WRITE(6,13) 1,2,3,4
      WRITE(6,13) T1,RL1,A1,B1
C = - RL1 /ALOG(1 -T1)
RM1 = -T1/((1-T1)*ALOG(1-T1))*RL1*A1/B1
C1 =C/(RL1 + B1)
L2 = IFIX(100*RM1)
C2 =0
      DO 2790 IXP1=1,L2+1
         X=IXP1-1
         IX=X
      Y(IXP1,1)=0
      F1=1.
C
C          COMPUTE (X-1)
C
IF( X.LT.2. )GO TO 2570
    DO 2560 K=1,IX-1
F1 = F1*K
2560 CONTINUE
2570 CONTINUE
Y (IXP1,IXP1)=1./F1
F1 = 1
IF( X.GT. 0. )GO TO 2630
S1 =(B1/(RL1 +B1))**A1
GO TO 2740
C
2630 CONTINUE
      S1=0.
      DO 2720 KK=1,IX
         KP1=KK+1
IF( IX .EQ. 1 )GO TO 2670
Y(IXP1,KP1)=Y(IXP1-1,KP1-1)/(IXP1-1) + Y(IXP1-1,KP1)
2670 CONTINUE
F1=1.
DO 2700 JJ=1,KK
      J=JJ-1
F1 = F1*(A1 + J)*C1
2700 CONTINUE
S1 = S1 + Y(IXP1,KP1)*F1
2720 CONTINUE
S1 = S1*(B1/(RL1 + B1))**A1*T1**X/X
2740 CONTINUE&
```

LPGFOR.S
LPG1

```
2740 C2 = C2 + S1
2750 IF( ID8 .LE.0 )GO TO 2780
2760 WRITE(6,13) X,S1,C2
2770 WRITE (8,13) X,S1,C2
2771 2780 CONTINUE
2780 IF( C2 .GT. .99  )GO TO 2800
2790 2790 CONTINUE
2795 2800 CONTINUE
2800 WRITE (8,13) -99,-99,-99
2810 RETURN
2815      END
```

LPGFOR.S
BSRCH

```
C
C      ***************************************************************
       SUBROUTINE LPG2
       COMMON/LPGPRM/T1,RL1,A1,B1,ID7,ID8
 WRITE(6,13) "SCALED POISSON"
  13 FORMAT(V)
C
 WRITE(6,13) "THETA =",T1,"LAMBDA =",RL1,"ALPHA =",A1,"BETA =",B1
 C = -RL1/ALOG(1-T1)
 C3= 1./(1.-T1)
 K1 = T1*C
 C4 = (B1/(K1+B1))**A1
 W = K1/(K1+B1)
 C2 = 0
     DO 3020 MM=1,1001
         M=MM-1
 P1 = 1
 IF( M.NE.0 )GO TO 3000
       P1 = C4
       GOTO 3040
C
 3000 CONTINUE
       DO 3020 II=1,M
          I=II-1
 P1 = P1*((A1+M-I-1)/(M-I)*W)
 3020 CONTINUE
 P1 = P1*C4
 3040 CONTINUE
 C2 = C2 + P1
   IF( ID8.LE.0 )GO TO 3080
 WRITE(6,13) M,C3*M,P1,C2
 WRITE (8,13) C3*M,P1,C2
 3080 CONTINUE
  IF( C2.GT..99  )GO TO 3100
 3090 CONTINUE
 3100 CONTINUE
 RETURN
       END
```

```
C       ******************************* *****************************
C
        SUBROUTINE LPG3
C
C                   NEGATIVE BINOMIAL APPROXIMATION.
C
        COMMON/LPGPRM/T1,RL1,A1,B1,ID7,ID8
 WRITE(6,13) "THIS PROGRAM COMPUTES  LPG USING SCALED BIN"
   13 FORMAT(V)
        WRITE(6,13)
 WRITE(6,13) "THETA =",T1,"LAMBDA =",RL1,"ALPHA =",A1,"BETA =",B1
        WRITE(6,13)
        WRITE(6,13) "NEG BINOMIAL PROBABILITIES"
        WRITE(6,13)
       WRITE(6,13) "            X            P            F"
        WRITE(6,13)
 C = 1/(1.-T1)
 RK1 = -T1*RL1/ALOG(1-T1)
 C4 = (B1/(RK1+B1))**A1
 W = RK1/(RK1+B1)
 C2 = 0
        DO 3550 MM=1,1001
         M=MM-1
 P1 = 1
 IF( M.NE.0 )GO TO 3370
 P1 = C4
 A = -1
  B = IFIX(C/2 +0.5)
 S2 = P1/(C/2+1)
 GOTO 3440
C
 3370 CONTINUE
        DO 3390 II=1,M
          I=II-1
 P1 = P1*((A1+M-I-1)/(M-I)*W)
 3390 CONTINUE
 P1 = P1*C4
 IA = IFIX((2*M-1)*C/2+0.5)
        RA=IA
 IB = IFIX((2*M+1)*C/2+0.5)
        RB=IB
 S2 = P1/C
 3440 CONTINUE
 RL3 = C2
        DO 3520 IX=IA+1,IB
          X=IX
 F3 = C2 + S2*(X-RA)
 F2 = F3 - RL3
 RL3 = F3
C
 WRITE(6,13) IX,F2,F3
 WRITE (8,13) IX,F2,F3
 3520 CONTINUE
 C2 = C2 + P1
 IF( C2 .GT. .99  )GO TO 3560
 3550 CONTINUE
 3560 CONTINUE
      WRITE (8,13) -99,-99,-99,"END OF NB"
 RETURN
        END
```

LPGFOR.S
LPG3

```
C***********************************************LPG4
      SUBROUTINE LPG4
C       LPG4.S
C                       COMPUTE REQUISITION SIZE R VS THETA TABLE
C
     DO 3670 I=1,100
         T1=I*.01
     B = -(1-T1)*ALOG(1-T1)
       R = T1/B
     WRITE(6,13) T1,R
 13 FORMAT(V)
 3670 CONTINUE
      RETURN
C       ****************************************************
C
C                            DEFINITIONS IN NAHMIAS' PROGRAM LPG1.S
C
C                 LAMB          RL1
C                 ALPH          A1
C                 BET           B1
C                 MEAN          RM1
C                 CNST          C1
C                 LIM           RL2
C                 CUM           C2
C                 FACT          F1
C                 SUM           S1
C                 KN            RK1
C                 PROD          P1
C                 CN            C3
C                 CONST         C4
C                 LAST          L3
C                 FX            F2
C                 FFX           F3
C                 SL           S2
 END
```

```
CATALOG/FILE DESCRIPTION=  INVR/SNBROP.S

10*#RUN *=;INVR/OBJ/SNBROP.O(BCD,NOGO)
20*SNBROP.S--CALCULATE SCALED NEGATIVE BINOMIAL CRITICAL POINTS
30C
40              SUBROUTINE SNBROP(AVEREQ,D1,ELT,CVLT,FCRIT,IROP)
50C
60        COMMON/IWT/IWT(20)
70C
80C              ESTIMATE LPG PARAMETERS
90C
100        IWT6 = IWT(6)
110C
120        IF(IWT6.GT.0) WRITE(6,13)AVEREQ,D1,ELT,CVLT,FCRIT
130C
140C              CHECK FOR NO DEMAND OR NO PENALTY FOR SHORTAGES CASE
150C                 IN THIS EVENT, SET RE-ORDER POINT TO ZERO
160C
170        IF(D1.LE.0.) GO TO 601
180        IF(FCRIT.LE.0.) GO TO 601
190C
200C
210        IF(ELT.LE..1)ELT=.1
220        IF(D1.LE.0.1)D1=0.1
230        IF(CVLT.LE.0.01)CVLT=0.01
240        CALL BSRCH(AVEREQ,Q)
250      T1=Q
260        S=CVLT*ELT
270        B=S*S/ELT
280        U1=D1*ELT
290C
300        B1=1./B
310        A1=B1*ELT
320        R1=-T1/((1.-T1)*ALOG(1.-T1))
330        RL1=U1/(ELT*R1)
340C
350        IF(IWT6.GT.0) WRITE(6,13)"        B1         A1",
360A         "        R1        RL1        C        RK1"
370    13  FORMAT(V)
380C
390C              ESTIMATE SNB PARAMETERS
400C
410        C=1./(1.-T1)
420         RK1=-T1*RL1/(ALOG(1.-T1))
430        IF(IWT6.GT.0) WRITE(6,63)B1,A1,R1,RL1,C,RK1
440     63 FORMAT(8F9.3)
450C
460C              (P(X=0.)-P0
470C
480         P0=(B1/(RK1+B1))**A1
490         W=RK1/(RK1+B1)
500C
```

```
C            SET UP INITIAL PROBABILITIES
C
      X=0.
      XL=0.
      FXL=0.
      PX=P0
       IF(PX.LE.0.) GO TO 701
      XN=C/2.
      FXN=P0
C
C             IS CUM>FCRIT
C             IF SO, EXIT LOOP.
C
C
C
      IF(INFO.GT.0)WRITE(6,13) "          X         PX",
&        "        FX          XL          FXL"
C
  220 CONTINUE
      IF((INFO.GT.0).AND.(X.LE.20.))WRITE(6,93)XN,PX,FXN,XL,FXL
   93 FORMAT(1X,2(F10.2,2F10.5))
      IF(FXN.GE.FCRIT) GO TO 410
      IF(FXN.GT. .999)GO TO 510
C
C             COMPUTE NEXT NEG. BIN PROB.
C
C
C             GENERATE SNB PROB TILL X*
C             INTERVAL IS FOUND
C
C
      X = X+1.
      XL=XN
      FXL=FXN
      XN=XN+C
       PX = PX * ( (A1+ X-1.)/ X   ) * W
          IF(PX.LE.0.) GO TO 701
      FXN=FXN+PX
      GO TO 220
C
C             INTERPOLATE TO FIND X*
C
  410 CONTINUE
      XCRIT=XL+(FCRIT-FXL)*(XN-XL)/(FXN -FXL)
C
C             ROUND TO GET FINAL VALUE
C
      IROP=(XCRIT+.5)
      RETURN
C
C              HIGH PROTECTION LIMIT. SET ROP TO UPPER LIMIT OF X.
C
   510 IROP=(XN+0.5)
        RETURN
C
C              DEMAND OR FCRIT IS ZERO. SET ROP=0
C
    601 CONTINUE
```

```
1090C
1100          IROP=0
1110           RETURN
1120C
1130C             PS =0, SO USE DEFAULT CALCULATION
1140C
1150   701 CONTINUE
1160             IROP = AVEREQ*DI*ELT*(1.+3.*CVLT)
1170             WRITE(6,723)IROP
1180   723 FORMAT(" *********SNBROP. PX=0, SSO SET ROP=",I8)
1190C
1200             RETURN
1210        END
1220C
1230        SUBROUTINE BSRCH(RO,Q)
1240C
1250            COMMON/IWT/IWT(20)
1260         ID7=IWT(6)
1270        ID8=IWT(6)
1280C        ---------------------------------AVE REQ SIZE EQUATION
1290C---------- FUNCTION FNR(Q) = -Q/( (1-Q)*ALOG(1-Q)   )
1300C
1310C             SEE 2182-2190 FOR FNR(Q) DEFINITION.
1320C
1330C
1340C        -----------------------------------
1350C                         BINARY SEARCH ROUTINE
1360C         -----------------------------------
1370C
1380C             SOLVE FOR THE VALUE OF Q WHICH GIVES AND
1390C               AVE. REQUISITION SIZE OF RO
1400C
1410C            SET UP END POINTS FOR SEARCH
1420C
1430      K= 0
1440     Q9=.999
1450     R9=FNR(Q9)
1460     Q1=.001
1470     R1=FNR(Q1)
1480        GOTO 2080
```

```
C
C               CHECK IF( RO .LE. R
C
 1940 CONTINUE
         IF( RO .GT. R )GO TO 2000
C
C                         RE-SET TOP OF INTERVAL
        R9=R
        Q9=Q
         GOTO 2080
C
 2000 CONTINUE
C
C                          RE-SET BOTTOM OF INTERVAL
        R1=R
         Q1=Q
          GOTO 2080
C
C                         HALVE THE INTERVAL, AND RE-EVALUATE THE FUNCTION
C
 2080 CONTINUE
         K=K+1
        Q=(Q1+Q9)/2
         R=FNR(Q)
C
C                         IF( R IS WITHIN .001 OF RO, )GO TO RETURN
C
        IF(IDB.GT.0) WRITE(6,23) K,RO,R,Q
  23     FORMAT(" ***BSRCH--K,RO,R,Q",I5,3F8.3)
         IF( ABS(R-RO) .LT. .001 )GO TO 2180
         IF( K .GT.25 )GO TO 2180
          GOTO 1940
 2180 CONTINUE
       RETURN
        END
        FUNCTION FNR(Q)
         FNR=-Q/((1-Q)*ALOG(1-Q))
       RETURN
       END
```

SNBROP.S

NEGATIVE BINOMIAL REQUISITION SIZE ROUTINES

DEMPAR

NEGBIN

NEGBN1

RANDEM

CATALOG/FILE DESCRIPTION= HEDG/DEMPR3.S

```
*#RUN=$HEDG/OBJ/DEMPR3.O(BCD,NOGO)
*DEMPR3.S
      SUBROUTINE DEMPAR(IDPER,IP4,IP5)
C***************************************************************************
         PARAMETER NOQQ=38
C***************************************************************************
      COMMON/IGBUG/IGBUG
      COMMON/ITIME/ITIME
      COMMON/ITWEEK/ITWEEK
       COMMON/ITMNTH/ITMNTH
       COMMON/ITQTR/ITQTR
       COMMON/ITYEAR/ITYEAR
      COMMON/NENTRY/NENTRY
      COMMON/NEEMAX/NEEMAX
       COMMON/NITEM/NITEM
       COMMON/PRI2F/PRI2F
       COMMON/NDENT/NDENT(1)
       COMMON/IDEMND/IDEMND(1,NQQQ)
       COMMON/IRETUR/IRETUR(1,NQQQ)
       COMMON/IREQ/IREQ(1,NQQQ)
       IF(IGBUG.NE.1) GO TO 25
       IY=ITIME/ITYEAR +1
       IQT=ITIME/ITQTR +1
       ITW=(ITIME-(IY-1)*ITYEAR)/ITWEEK +1
       WRITE(6,103)IY,IQT,ITW
  103 FORMAT(" DEMPAR-----",5X,"YEAR NO.",I3,5X,"QTR NO.",I3,5X,
&       "WEEK NO.",I3)
   25 CONTINUE
C
C        INCREMENT PERIOD COUNTERS
C
       IDPER=IDPER+1
       DO 100 N=1,NITEM
       IQTY=IDEMND(N,IDPER)
       IRET=IRETUR(N,IDPER)
       IRQ=IREQ(N,IDPER)
       IF(IGBUG.EQ.1)WRITE(6,8013)N,NDENT(N),IDPER,IQTY,IRET,IRQ
 8013      FORMAT(4X,'****DEMPAR--N=',I5,'  NDENT=',I5,' IDPER=',I5,
&             ' IDEMND=',I5,' IRETRN=',I5,' IREQ=',I5)
       IF(NDENT(N).LE.0) GO TO 100
C
C             GENERATE SERVICEABLE RETURNS, IF ANY
C
       IF(IRET.LE.0)GO TO 30
       IT=ITIME+ITMNTH
       CALL ENTER(IT,4,N,IRET,0)
   30      CONTINUE
C
C       REFLECT      REQUISTION FOR ITEM N
C
C
C           SET REQUISTION COUNTER KNTREQ TO ZERO
C
```

DEMPR3.S
DEMPAR

```
      KNTREQ=0
       IF(IQTY.LE.0) GO TO 100
       ITOT=0
C
C               OBTAIN A REQUISITION SIZE IR BY MONTE CARLO
C
   50       R=RANDEM(.2)
      CALL NEGBIN(R,IRO)
       ITEMP=ITOT+IRO
       IR=IRO
       IF(ITEMP.LE.IQTY) GO TO 60
C
C               LIMIT LAST REQUISITION SO THAT TOTAL DEMAND IN QTR
C                  EQUALS IQTY.
C
       IR=IQTY-ITOT
   60       CONTINUE
C
      KNTREQ= KNTREQ + 1
C
C            IF (NFEMAX-50) REQUISITIONS HAVE BEEI
C             GENERATED, PUT ALL REMAINING DEMAND ON THIS REQUISITON.
C
C
       IF( KNTREQ .LT. (NFEMAX - 50) ) GO TO 80
C
      IR = IQTY-ITOT
       WRITE(6,73)ITIME,IR,IQTY
   73 FORMAT(4X,'****DEMPAR--ITIME=',I8,'   F.E.L ',
     &          ' OVERFLOW REQ=',I5,'  OF ',I5)
           WRITE(6,8013)N,NDENT(N),IDPER,IQTY,IRET,IRQ
C
C             DETERMINE PRIORITY OF THIS REQUISITION
C              ASSUMING 50% ARE PRIORITY 1
C
   80 CONTINUE
       IPR=1
       IF(RANDEM(.2).LE. PRI2F) IPR=2
C
C
C              DETERMINE ARRIVAL TIME FOR THIS REQUISITION
C
C       LIMIT ARRIVAL TIME TO NO LATER THAN .5 WEEKS
C       BEFORE THE END OF THE QUARTER
C
       LM=ITOTR-0.5*IWEEK
       IT=ITIME+RANDEM(.2)*FLOAT(LM)
       IF(IDBUG.EQ.1)WRITE(6,8023)IT,N,R,IRO,IR,IPR
 8023       FORMAT(4X,'****DEMPAR--IT=',I10,'  N=',I4,'  R=',F5.3,
     &          '  INITAL REQ-SIZE=',I4,'  FINAL REQ-SIZE=',I4,
     &          '  PRIORITY=',I4)
C
C              PUT REQUISITION OF F.E.L.
C
```

```
      CALL ENTR (IT,I,N,IR,IRQ)
      ITOT=ITOT+IR
      IF(ITOT.LT.IQTY) GO TO 50
C
C********CREATE NEXT DEMPAR EVENT
C
 100        CONTINUE
      IT=ITIME+ITOTR
      CALL ENTER(IT,12,IDPER,0,0)
      RETURN
      END
C            NOTE * GETREQ IS NO LONGER USED.
C                  RATHER, NEGBNI GENERATES NEGBIN REQ SIZES.
C
```

```
1160        SUBROUTINE GETREQ(R,IQTY,IR)
1170C               THIS ROUTINE DETERMINES A REQUISITION SIZE
1180C                  CORRESPONDING TO A CUM PROB. OF R.
1190C
1200       DIMENSION IUCAT(8),CPROB(10,8),IRQ(10,8)
1210       DATA IUCAT/1,3,10,31,100,316,1000,3162/
1220       DATA CPROB/10*1.,
1230&                .40,.82,8*1.0,
1240&                .27,.52,.65,.73,.86,.91,.92,.95,.96,1.0,
1250&                .23,.53,.73,.78,.82,.90,.92,.97,.99,1.00,
1260&                .06,.38,.69,.73,.81,.85,.89,.93,.98,1.00,
1270&                .015,.23,.42,.50,.65,.70,.76,.92,.97,1.00,
1280&                .012,.17,.31,.37,.48,.55,.60,.78,.87,1.00,
1290&                .001,.03,.32,.50,.60,.66,.86,.96,1.00,1.00/
1300       DATA IRQ/10*1,
1310&                 1, 2, 3, 7*3,
1320&                 1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
1330&                 1, 4, 6, 8,10,13,15,20,25, 30,
1340&                 1, 5,10,15,20,25, 30,40,70,100,
1350&                 1, 5,10,15,20,25,30,80,150,300,
1360&                 1, 5,10,15,20,25,30,80,200,600,
1370&                 1,20,40,60,80,100,200,300,600,700/
1380C
1390C
1400C              ESTABLISH DEMAND RATE CATEGORY
1410C
1420       DO 10 I=1,8
1430       IM=I
1440       M=IUCAT(I)
1450       IF(IQTY.LE.M) GO TO 30
1460   10      CONTINUE
1470C
1480C              DETERMINE REQUISITION SIZE
1490C
1500   30      CONTINUE
1510       DO 40 I=1,10
1520       IF(CPROB(I,IM).GE.R) GO TO 60
1530   40      CONTINUE
1540   60      CONTINUE
1550C
1560C              SET REQUISITION SIZE
1570C
1580       IF(I.GT.1)GO TO 80
1590       IR=1
1600       RETURN
1610   80      CONTINUE
1620       IF(I.LT.10)GO TO 100
1630       IR=IRQ(10,IM)
1640       RETURN
1650  100      CONTINUE
1660       DPROB=CPROB(I,IM)-CPROB(I-1,IM)
1670       RPROB=R-CPROB(I-1,IM)
1680       DRQ=IRQ(I,IM)-IRQ(I-1,IM)
1690       IR=IRQ(I-1,IM)+IFIX(RPROB*DRQ/DPROB+.5)
1700       RETURN
1710       END
```

DEMPR3.S
GETREQ

```
C                    COMPUTE GRAND MEAN  X9
C
   50 CONTINUE
      X9=T9/C9
C
C              IF THERE IS ONLY 1 OBSERVATION, ASSUME R=.0439
C
      IF (NPT .GT.1) GO TO 59
C
C              USE AVERAGE R9 VALUE,
C
   51 R9=.0439
      P9=(X9-1)/R9
      Q9=P9+1
C
C              IF P9 =0, USE AVE ITEM VALUES
C
      IF(P9.LE.0.) GO TO 41
      GO TO 170
C
C                   COMPUTE VAR(X) ESTIMATE
C
   59 CONTINUE
      V=0.
      DO 60 I=1,NPT
      V=V+C(I)*(X(I)-X9)**2
   60 CONTINUE
      V=V/(FLOAT(NPT-1))
      S=SQRT(V)
C
C              IF VARIANCE > (X9-1), CONTINUE TO 100, OTHERWISE,
C                  A NEGATIVE BINOMIAL MODEL DOESN'T FIT.
C
      IF(IWT(3).GE.2) WRITE(6,73)NPT,X9,V,S,S/X9,(X(I),I=1,NPT)
   73 FORMAT(T20,'NPT=',T30,I10/
&             T20,'MEAN=',T30,F13.2/
&             T20,'VAR=',T30,F13.2/
&             T20,'STD DEV=',T30,F13.2/
&             T20,'COEF OF V=',T30,F13.2/
&             T20,'X(I)=',T30,20F6.2)
      IF(V.GT.(X9-1))GO TO 100
     IF(IWT(3).GE.1)
&         WRITE(6,63)NPT,X9,V,S,S/X9,(X(I),I=1,NPT)
   63 FORMAT(4('****'),'NEGBIN. VAR(X) <= MEAN.',
&          T20,'NPT,MEAN,VAR,STD DEV,C.OF.V =',I4,4F13.2/
&          T20,'X(I)=',20F6.2)
C
C              CHECK FOR CONSTANT REQ SIZE. IF NOT , GO TO 51
C
      IF( (V.GT. 0.05) .OR. (NPT.LT.4) ) GO TO 51
C
C              USE CONSTANT REQ SIZE
C
```

DEMPR3.S
NEGBIN.S

```
3340C
3350       FDIF=F-PREQS(NREQS)
3360       IF(I.LE.25) GO TO 184
3370       IF(FDIF.LT.0.01)GO TO 185
3380C
3390C                     COUNT IT
3400C
3410   184 NREQS=NREQS+1
3420       IF(NREQS.GT.100) GO TO 200
3430C
3440       IREQS(NREQS)=I+1
3450       PREQS(NREQS)=F
3460   185 CONTINUE
3470C
3480       IF(F.GT.0.999)GO TO 200
3490C
3500   170 CONTINUE
3510   200 CONTINUE
3520C
3530C                     PUT LAST POINT INTO PREQS(I) TABLE
3540C
3550       NREQS=NREQS +1
3560       IF(NREQS.GT.100) NREQS=100
3570       IREQS(NREQS)=I+1
3580       PREQS(NREQS)=1.0
3590C
3600C                     IF IWT(3).GE.2, WRITE PROBABILITY ARRAY
3610C
3620    210  IF(IWT(3).LT.2)GO TO 250
3630       WRITE(6,213)
3640   213 FORMAT(//T20,'REQUISITION SIZE C.D.F.'//
3650&                   '      I      R     P(X<=R)'/)
3660       DO 230 I=1,NREQS
3670C
3680       WRITE(6,223)I,IREQS(I),PREQS(I)
3690   223 FORMAT(I4,I6,F11.4)
3700   230 CONTINUE
3710    250  CONTINUE
3720         IF(IWT(3).GE.2)WRITE(6,323)
3730    323  FORMAT(//T20,'INITIAL REQUISITION COUNTS'//
3740&                     T7,'QTR',T15,'UNITS',T27,'REQ'//)
3750C
3760C                USE MONTE CARLO TO SET FIRST NDHIS
3770C                REQ COUNTS CONSISTENT WITH THE NEG BIM ESTIMATES
3780C
3790       DO 380 I=1,NDHIS
3800         IQTY=IDEMND(I)
3810       KNT=0
3820       ITOT=0
3830       IF(IQTY.LE.0)GO TO 370
3840C
3850C                GENERATE REQUISITIONS TELL TOTAL UNITS=>IQTY.
3860C
```

DEMPR3.S
NEGBIN.S

```
      NREQS = 1
      PREQS(1) = 1.00
      IREQS(1) = X9 + .5
      GO TO 210
C
C
C
C                   COMPUTE NEGATIVE BINOMIAL PARAMETERS
C
  100 Q9=V/(X9-1)
      P9=Q9-1
      R9=(X9-1)/P9
      R2=(X9-1)**2/(V-X9+1)
C
C             LET
C                   IREQS(I)=ITH REQUISITION SIZE
C                   PREQS(I)=PROBABILITY REQUISITION SIZE <=IREQS(I)
C                   NREQS=NUMBER OF POINTS IN REQUISITION SIZE C.D.F. APP
C
C                   R9,P9,Q9=NEGATIVE BINOMIAL PARAMETERS
C
C                    COMPUTE NEGATIVE BINOMIAL PROBABILITIES
C                         USING R9 AND Q9
C
C                   INITIALIZE VARIABLES
  170 CONTINUE
      IF(IWT(3).GE.2)WRITE(6,173)R9,P9,Q9
  173 FORMAT(///'NEG. BIN PARAMETERS'/
     &           '   R9=',F8.3,'   P9=',F8.3,'   Q9=',F8.3)
C
      Y=0.
      P=1/(Q9**R9)
      F=P
        AVE= 1-P
      P8=P9/Q9
      IREQS(1)=1
      PREQS(1)=F
      NREQS=1
C
C                   COMPUTE NEGATIVE BINOMIAL PROBABILITIES
C
C              MAX REQUISITION SIZE ALLOWED = 1000 UNITS
      DO 190 I=1,1000
      IPRNT=1
      IF(I.GT.20)IPRNT=0
      IF(MOD(I,10).EQ.0)IPRNT=1
      IF( (IWT(3).GE.3).AND.(IPRNT.EQ.1))WRITE(6,183)I,Y,P,F,AVE
  183 FORMAT(' I,Y,P,F,AVE=',I10,F10.0,3F10.4)
      Y=I-1
      P=((Y+R9)/(Y+1))*P8*P
      F=F+P
        AVE=AVE + (1-F)
C
C                   RECORD APPROXIMATION TO F(X) AT LEAST EVERY .01 INCR
```

DEMPR3.S
NEGBIN.S

```
  350  R=RANDEA(.2)
       CALL NEGBNI(R,IR)
       ITOT=ITOT+IR
       KNT=KNT+1
       IF(ITOT.LT.IOTY)GO TO 350
C
C                SET NUMBER OF REQUISITIONS EQUAL TO KNT.
C
  370  IREQ(I)=KNT
C
C               WRITE DEBUG MESSAGE.
C
       IF(INT(3).GE.2)WRITE(6,373)I,IDEMND(I),IREQ(I)
  373  FORMAT (3I10)
C
  380  CONTINUE
       RETURN
       END
```

DEMPR3.S
NEGBIN.S

```
      SUBROUTINE NEGBN1(R,IR)
C*****************************************************************************
      PARAMETER NQQQ=38
C**************************************************************************
C
      COMMON/INT/INT(20)
      COMMON/IDEMND/IDEMND(NQQQ)
      COMMON/IREQ/IREQ(NQQQ)
      DIMENSION B(10),C(10),D(10)
      COMMON/NEGBND/ NREQS, IREQS(100),PREQS(100)
C                    GIVEN R, A U(0,1) PSEUDO RANDOM NUMBER,
C             .           DETERMINE THE CORRESPONDING REQUISITION SIZE IR.
C
      DO420 I=1, NREQS
      IF(R.LE.PREQS(I))GO TO 430
  420 CONTINUE
  430 CONTINUE
C
C             .     SET THE REQUISITION SIZE IR
C
      IF(I.GT.1)GO TO 460
      IR=IREQS(I)
      RETURN
C
  460 CONTINUE
      IF (I.LT. NREQS)GO TO 480
      IR=IREQS(NREQS)
      RETURN
C
  480 CONTINUE
      IM1=I-1
      IRI=IREQS(I)
      IRIM1=IREQS(IM1)
      IRDIF=IRI-IRIM1
      IF(IRDIF.GT.1)GO TO 510
      IR=IRI
      RETURN
C
C                   INTERPOLATE TO DETERMINE REQUISITION
C
  510 CONTINUE
      DPROB=PREQS(I)-PREQS(IM1)
      RPROB=R-PREQS(IM1)
      IR=IRIM1+IFIX((RPROB*IRDIF/DPROB)+0.5)
      RETURN
C
      END
```

DEMPR3.S
NEGBN1

```
4530C------------------RANDEM--U(0,1) RANDOM NUMBER GENERATOR.
4540C             RANDEM--U(0,1)  RANDOM NUMBER GENERATOR.
4560      FUNCTION RANDEM(X)
4570C             A CALL WITH X < 0. INITIALIZES THE RANDOM NUMBER STREAM.
4580      IF(X) 10,20,20
4590   20 RN=RHO*RANDEM
4600      RN1=AMOD(RN,BN)
4605      RANDEM=RN1/BN
4610      RETURN
4620   10 RHO=7.0**13
4630      BN=10.0**10
4640      RANDEM=-X
4650      GO TO 20
4660      END
```

```
CATALOG/FILE DESCRIPTION=   HEDG/INITM3.S

10*#RUN=$HEDG/OBJ/INITM3.O(BCD,NOGO)
20*INITM3.S
30         SUBROUTINE INITEM
40C-------------------------------------------------------------------
50         PARAMETER NOQQ=38
60C-------------------------------------------------------------------
70        CHARACTER ALC,FSN,UM,NOUN,MGTCD
80        COMMON/FSN/ALC,FSN(4),UM,NOUN(2),MGTCD(4),IOH,IOR,IPPI,IPPPR
90         COMMON/ISLE/ISLE
100          COMMON/IDEBUG/IDEBUG
110          COMMON/ITDAY /ITDAY
120          COMMON/ITMBTH/ITMBTH
130          COMMON/NITEM/NITEM
140        COMMON/NDEM/NDEM
150        COMMON/NDHIS/NDHIS
160        COMMON/INEQ/INEQ
170        COMMON/INTYPE/INTYPE
180        COMMON/IFBUG/IFBUG
190          COMMON/IBOP/IBOPOH(3),IBOPOR(3)
200          COMMON/IDEMND/IDEMND(1,NOQQ)
210          COMMON/RMREQS/RMREQS(1)
220          COMMON/INVACT/INVACT(1)
230          COMMON/NORDPT/IORDPT(1)
240          COMMON/NDEMAC/NDEMAC(1)
250        COMMON/NRETAC/NRETAC(1)
260        COMMON/NREQAC/NREQAC(1)
270        COMMON/NDEMND/NDEMND(1,NOQQ)
280        COMMON/NRETUR/NRETUR(1,NOQQ)
290        COMMON/NREQ/NREQ(1,NOQQ)
300          COMMON/NDENT/NDENT(1)
310          COMMON/INVDUE/INVDUE(1)
320          COMMON/NBOPT/NBOPT(1)
330          COMMON/NBOTU/NBOTU(1)
340        COMMON/NBOIU/NBOIU(1)
350        COMMON/NBOIR/NBOIR(1)
360        COMMON/NBOTR/NBOTR(1)
370          COMMON/REQSIZ/REQSIZ(1)
380          COMMON/REQMAD/REQMAD(1)
390          COMMON/LTPROD/LTPROD(1)
400          COMMON/LTADM/LTADM(1)
410          COMMON/UCOST/UCOST(1)
420          COMMON/ADR/ADR(1)
430          COMMON/ISUL/ISUL(1)
440        COMMON/IREQ  /IREQ(1,NOQQ)
450        COMMON/IRETUR/IRETUR(1,NOQQ)
460          COMMON/IRL/IRL(1)
470          COMMON/ITL/ITL(1)
480          COMMON/IROL/IROL(1)
490          COMMON/IROTY/IROTY(1)
500          COMMON/RMTBR /RMTBR(1)
510          COMMON/RMEAN/RMEAN(1)
520          COMMON/RTREND/RTREND(1)
530          COMMON/RMAD  /RMAD(1)
540          COMMON/RERSUM/RERSUM(1)
```

INITM3.S

```
      COMMON/KNT    /KNT(1)
      COMMON/GROO/GROO(3)
      COMMON/GROOF/GROOF(3)
     COMMON/ICDFOR/ICDFOR
C
      ENTRY INITM1
C
C             SET NUMBER OF PERIODS OF DATA FROM
C
     IDPER=NDEM
C
C       READ DEMAND DATA FOR ITEM N FROM LOGICAL UNIT LR
     LR=INLU
      IKNT=IKNT+NITEM
      DO 100 N=1,NITEM
   10 CONTINUE
C
C             READ ITEM DATA INPUT FROM FILE LR
C
     IF(INTYPE.EQ.2) GO TO 20
C
C             READ BCD INPUT
C
C
     READ(LR,8000,END=200) ALC,FSN,UM,UCOST(N),NOUN,MGTCD,IOH,IOR,
&             LTADM(N),LTPROD(N),IPPL,IPPPR
C
     RIPPPR=FLOAT(IPPPR)/100.
     IF(IEBUG.EQ.1)WRITE(6,8010)IKNT,ALC,FSN,UM,UCOST(N),NOUN,MGTCD,
&             IOH,IOR,LTADM(N),LTPROD(N),IPPL,RIPPPR
     IF(IEBUG.EQ.1)WRITE(6,8015)(I,I=1,10)
C
C             READ DEMAND,RETURNS, AND REQ-FREQUENCY
C
     READ(LR,8000)(IDEMND(N,J),J=1,IDPER)
     IF(IEBUG.EQ.1)WRITE(6,8020)(IDEMND(N,J),J=1,IDPER)
     READ(LR,8000)(IRETUR(N,J),J=1,IDPER)
     IF(IEBUG.EQ.1)WRITE(6,8030)(IRETUR(N,J),J=1,IDPER)
     READ(LR,8000)(IREQ(N,J),J=1,IDPER)
     IF(IEBUG.EQ.1)WRITE(6,8040)(IREQ(N,J),J=1,IDPER)
     GO TO 50
C
C             READ BINARY DATA
C
   20      CONTINUE
```

```
       IKNT=IKNT+1
       READ(L,2)ALC,FSN,IR,UM,UCOST(N),NOUN,MGTCD,IOH,IOR,
&            LTADA(N),LTPROD(N),IPPL,XIPPPR
       IF(IFBUG.EQ.1)WRITE(6,8010)IKNT,ALC,FSN,UM,UCOST(N),NOUN,
&            MGTCD,IOH,IOR,LTADA(N),LTPROD(N),IPPL,XIPPPR
       READ(L,2)ALC,FSN,IR,IDEMND
       IF(IFBUG.EQ.1)WRITE(6,8020)(IDEMND(N,J),J=1,IDPER)
       READ(L,2)ALC,FSN,IR,IRETUR
       IF(IFBUG.EQ.1)WRITE(6,8030)(IRETUR(N,J),J=1,IDPER)
       READ(L,2)ALC,FSN,IR,IREQ
       IF(IFBUG.EQ.1)WRITE(6,8040)(IREQ(N,J),J=1,IDPER)
C            WRITE INPUT DATA TO FILE 09
C
C
C
 8000        FORMAT(V)
 8010        FORMAT(/I5,1X,A2,1X,A2,A4,A6,A3,1X,A2,F11.2,1X,A6,A4,
&             T53,2A1,A4,A2,2X,2I7,3I6,F5.2)
 8015        FORMAT(T21,10I10)
 8020        FORMAT('IDEMND  DEMAND/QTR',(T21,10I10))
 8030        FORMAT('IRETUR  RETURN/QTR',(T21,10I10))
 8040        FORMAT('IREQ    REQ    /QTR',(T21,10I10))
C
   50        CONTINUE
C
C             DETERMINE PARAMETERS FOR NEGATIVE BINOMIAL
C                 REQUISITION SIZE GENERATION
C
       CALL NEGBIN
C
  100 CONTINUE
        RETURN
C
C             INITIALIZE ITEM ARRAYS FOR A NEW SIM. RUN
C
       ENTRY INITM2
C
        DO 210 N=1,NITEM
C
C                  INDICATE ITEM HAS NDHIS PERIODS OF DEMAND HISTORY
C
        NDENT(N)=NDHIS
C
C       ZERO DEMAND HISTORY RECORDS
       NRETAC(N)=0
        NDEMAC(N)=0
       NREQAC(N)=0
C
```

INITM3.S

```
1480C            LOAD DEMAND HISTORY ARRAYS
1490C
1500      KK=NOHIS
1510      DO 1  I=1,NOHIS
1520      NDEMND(N,KK)=IDEMND(N,I)
1530      NRETUR(N,KK)=IRETUR(N,I)
1540      NREQ(N,KK)=IREQ(N,I)
1550      KK=KK-1
1560 1      CONTINUE
1570C
1580C       SET INVENTORY DUE-IN TO ZERO
1590   70 INVDUE(N)=0
1600      NORDPT(N)=0
1610C
1620C      SET INITIAL BACKORDER COUNTERS TO ZERO
1630C
1640       NBOFO(N)=0
1650      NBOIO(N)=0
1660      NBOIR(N)=0
1670      NBOTR(N)=0
1680       NBOPT(N)=0
1690C
1700C            COMPUTE DEMAND RATES
1710C
1720      NN=N
1730      CALL FOR576(NN)
1740C
1750C            SET INITIAL INVENTORY ON-HAND EQUAL TO LEAD-TIME DEMAND
1760C
1770      RLT=(LTADM(N)+LTPROD(N))*ADR(N)/12.
1780      INVACT(N) = RLT +0.5
1790C
1800C
1810   210 CONTINUE
1820C
1830C         INITIALIZE GROSS ON-HAND AND ON-ORDER STATISTICS
1840C
1850  110 CONTINUE
1860       DO 130 N=1,NITEM
1870       IF(NDENT(N).LT.0) GO TO 130
1880       IF(INVACT(N).LE.0) GO TO      120
1890       IBOPOH(1)=IBOPOH(1)+1
1900       IBOPOH(2)=IBOPOH(2)+INVACT(N)
1910       IBOPOH(3)=IBOPOH(3)+IFIX(UCOST(N)*FLOAT(INVACT(N)))
1920  120 CONTINUE
1930       IF(INVDUE(N).LE.0) GO TO      130
1940      IBOPOR(1)=IBOPOR(1)+1
1950       IBOPOR(2)=IBOPOR(2)+INVDUE(N)
1960       IBOPOR(3)=IBOPOR(3)+IFIX(UCOST(N)*FLOAT(INVDUE(N)))
1970  130 CONTINUE
1980  140 CONTINUE
1990       RETURN
2000  200 WRITE(6,8200)
2010 8200 FORMAT(1H1,////20X,'END OF FILE READING...................'////
2020       CALL OUT
2030      CALL OUTCST
2040       CALL PLOTR
2050       STOP
2060       END
```

INITM3.S

```
*#RUN=;HEDG/OBJ/INORD3.O(BCD,NOGO)
*INORD3.S
      SUBROUTINE INORD(N,IOH,INIOR,IBO,IROL,IRQTY)
C*************************************************************
       PARAMETER NQQQ=38
C*************************************************************
C
C       THIS SUBROUTINE DETERMINES THE RECEIPT TIME FOR ASSETS
C       ON-ORDER AT THE BEGINNING OF THE SIMULATION.  IT IS
C       CALLED AT THE BEGINNING OF EACH REPLICATION AND ITS
C       OUTPUT IS STORED IN TWO TABLES (IDUE & IORQ) TO BE USED
C       FOR EACH SHORTAGE FACTOR WITHIN THE REPLICATION.  WHEN
C       THE ON-ORDER QUANTITY IS ONE AND A HALF TIMES THE
C       EOQ OR LESS, THE TOTAL ON-ORDER ASSETS ARE ASSUMED
C       TO BE THE RESULT OF ONE PROCUREMENT ACTION AND
C       ALL DUE IN AT ONE TIME.  WHEN ON-ORDER ASSEST
C       EXCEED ONE AND A HALF EOQ'S, THIS SUBROUTINE THEN
C       COMPUTES A DJE-IN DATE FOR EACH REMAINING EOQ SO
C       LONG AS THE COMPUTED DUE-IN DATE IS GREATER THAN
C       IFIVE (IE,500).  WHEN THE DUE-IN DATE IS LESS
C       THAN IFIVE, THE REMAINING ON ORDER IS SET AS DUE-IN
C       AT IFIVE.
C
      COMMON/IDBUG/IDBUG
      COMMON/IEBUG/IEBUG
     COMMON/NDHIS/NDHIS
      COMMON/NDEMND/NDEMND(1,NQQQ)
      COMMON/NRETUR/NRETUR(1,NQQQ)
      COMMON/ITDAY/ITDAY
      COMMON/ITMNTH/ITMNTH
      COMMON/LTPROD/LTPROD(1)
      COMMON/LTADM/LTADM(1)
      COMMON/IDUE/IDUE(NQQQ)
      COMMON/IORQ/IORQ(NQQQ)
      COMMON/JCTR/JCTR
C
C     WRITE DEBUG MESSAGE
C
      IF(IEBUG.EQ.1) WRITE(6,901)IOH,INIOR,IBO,IROL,IRQTY
  901 FORMAT("..INORD.S.. IOH=",I6,"  INIOR=",I6,"  IBO=",I6,"  IROL=",
     &I6,"  IRQTY=",I6)
C
C
C     ZERO ON-ORDER ARRAYS
C
      DO 5 I=1,NQQQ
      IDUE(I)=0
      IORQ(I)=0
    5 CONTINUE
C-------------------------------------------------------------
C                                    LOGIC TO TEST LEAD-TIME SENSITIVITY
C
```

```
      IOPT=1
      IF(IOPT.NE.2) GO TO 7
C
C            SET ALL ON-ORDER STOCK DUE-IN ONE LEAD-TIME IN
C                THE FUTURE.
C            THIS IS A VERY PESSIMISTIC ASSUMPTION.
C
      JCTR=0
      IF(INIOR.LE.0) RETURN
      JCTR=1
      IDUE(1)=(LTADM(N)+LTPROD(N))*ITMNTH
      IORQ(1)=INIOR
       RETURN
    7 CONTINUE
C-------------------------------------------END OF SENSITIVITY LOGIC
C
C       LET
C
C           ISTKOBJ=STOCKING OBJECTIVE
C           NETASSTS=NET AVAILABLE STOCK
C           IFIVE=CLOCK TIME 5 DAYS INTO SIMULATION
C           CHECKIOR=MAX SIZE FOR ONE REQUISITION
C           IHOLDIOR=WORKING VAR. FOR REMAINING ON-ORDER STOCK
C          NREOQSOO=MAX REQUISITION ON-ORDER (INTEGER)
       IOR=INIOR
       ISTKOBJ=IROL+IRQTY
       NETASSTS=IOH+IOR-IBO
       IFIVE=5*ITDAY
       CHECKIOR=FLOAT(IRQTY)*1.5
       IHOLDIOR=IOR
C
C            IF NREOQSOO >  NDHIS , SET EQUAL TO  NDHIS  TO AVOID ARRAY OV
C
       KQTR=1
C                    COMPUTE MAXIMUM NO OF OUTSTANDING ORDERS
      NREOQSOO=IOR/IRQTY +1
       IF(NREOQSOO.GT. NDHIS )NREOQSOO= NDHIS
       NDMDS=0
       DO 50 J=1,NREOQSOO
       DO 10 K=KQTR,8
       NDMDS=NDMDS+NDEMND(N,K)-NRETUR(N,K)
       IAVAIL=NETASSTS+NDMDS
       IF(IEBUG.EQ.1) WRITE(6,903)J,K,NDMDS,IAVAIL
  903 FORMAT("      ORDER NO=",I2,"   QTR=",I2,"   NET DEMAND=",I6,
&      "   IAVAIL=",I6)
       IF(IAVAIL.GT.ISTKOBJ)GOTO 20
   10 CONTINUE
C
      CALL INTIME(N,RLT)
      XMNTH=(RLT-24.)
      IDUE(J)=XMNTH*FLOAT(ITMNTH)
      IF(IDUE(J).LT.IFIVE)IDUE(J)=IFIVE
      GOTO 30
C
C           COMPUTE RANDOM TIME WITHIN QTR FOR ORDER RECEIPT.
C           SET IDUE(J)=DUE-IN TIME FOR ORDER NO.J.
C
   20 T=RANDU(.2)
      KQTR=K+1
      ORDQTR=K
     CALL INTIME(N,RLT)
     XMNTH = RLT + 3.*(T-ORDQTR)
      XIDUE=XMNTH*FLOAT(ITMNTH)
      IDUE(J)=IFIX(XIDUE)
C
C           LIMIT DUE-N TIME TO AT LEAST 5 DAYS IN FUTURE.
C
      IF(IDUE(J).GT.IFIVE)GOTO 25
      IDUE(J)=IFIVE
      GOTO 30
C
C          SET IORQ(J)=QUANTITY FOR ORDER J.
C
   25 IF(IOR.LE.CHECKIOR)GOTO 30
      IORQ(J)=IRQTY
       GO TO 35
   30 CONTINUE
      IORQ(J)=IOR
   35 CONTINUE
      IF(J.EQ. NDHIS )IORQ(J)=IHOLDIOR
      IF(IEBUG.EQ.1) WRITE(6,904)T,XMNTH,IDUE(J),IORQ(J),IHOLDIOR
  904 FORMAT("        T=",F10.8,"  XMNTH=",F10.6,"  IDUE(J)=",I7,
&"  IORQ(J)=",I7,"   ON-ORDER BEFORE PLACING THIS ORDER =",I7)
C
C          UPDATE COUNTER VARIABLES
C
      IF(IHOLDIOR-IORQ(J))200,100,40
   40 IHOLDIOR=IHOLDIOR-IORQ(J)
      ISTKOBJ=ISTKOBJ+IRQTY
      IOR=IHOLDIOR
   50 CONTINUE
 100  JCTR=J
      RETURN
C
C           PRINT ERROR MESSAGE
C
 200     CONTINUE
         WRITE(6,213)
  213    FORMAT("*****INORD---ERROR--")
         IDBUG=1
         RETURN
      END
```

INORD3.S

```
*INTIME.S
C
      SUBROUTINE INTIME(N,RLT)
C           THIS ROUTINE COMPUTES THE ACTUAL LEADTIME, RLT,
C       FOR A GIVEN REPLENISHMENT ORDER FOR ITEM N,
C
C           DEFINITIONS
C          CP(I) = PROB( ACT LT TO PRED LT <= RATIO(I))
C            RATIO(I)= RATIO OF ACTUAL LEADTIME TO PREDICTED LEADTIME
C                       ASSOCIATED WITH CLASS I
C
      COMMON/IWT/IWT(20)
      COMMON/LTPROD/LTPROD(1)
      COMMON/LTADM /LTADM(1)
      COMMON/IRNDLT/IRNDLT
      COMMON/ITIME/ITIME
C
      DIMENSION CP(10),RATIO(10)
C
C             THE CP(I) DATA IS FOR A GAMMA PDF WITH MEAN = 1
C               AND COEF. OF VAR. = .353. > 1/√ALPHA
C                HENCE, ALPHA = 8, BETA = 1/8.
C          THIS IS THE MEDIAN COEF OF VAR. FOR THE 62 ITEMS
C          REPORTED IN APPENDEX D OF HAYYA(1980).
C
      DATA CP/.000,.077,.255,.490,.700,
&             .844,.927,.969,.987,1.00/
      DATA RATIO/0.3,.5,.7,     .9, 1.10,
&              1.3, 1.5, 1.7,  1.9, 2.70/
C
      IDBUG=IWT(1)
C
C           IF IRNDLT = 1, MONTE CARLO TO DETERMINE LEADTIME.
C             OTHERWISE, SET LEADTIME = PREDICTED VALUE.
C
      IF(IRNDLT.EQ.1) GO TO 17
C
C           SET LEADTIME TO PREDICTED VALUE
C
      RLT = LTPROD(N) + LTADM(N)
       RETURN
C
C           OBTAIN A U(0,1) RANDOM NUMBER R
C
   17 CONTINUE
      R=RANDU(.2)
C
C           FIND PROB. CLASS THAT INCLUDES R
C
      DO 20 I=2,10
       IF(R.LE.CP(I))GO TO 40
   20 CONTINUE
   40 CONTINUE
C
C              INTERPOLATE FOR EXACT RATIO V
C
      IL=I-1
      RDELT=RATIO(I)-RATIO(IL)
      CPDELT=CP(I)-CP(IL)
C
      V=RATIO(IL) + (RDELT/CPDELT)*(R-CP(IL))
C
      LEADTM= LTPROD(N) + LTADM(N)
      RLT=V*LEADTM
C
      IF(IDBUG.EQ.1)WRITE(6,113)ITIME,N,R,LEADTM,V,RLT
  113 FORMAT(" ----INTIME-ITIME=",I8,"  N=",I4,"  R=",F4.2,
&               " PLANNED LEADTIME=",I3," V=",F4.2,
&               " RANDOM LEADTIME (MNTHS)=",F5.2)
C
      IF(RLT.LT..1) RLT=.1
C
      RETURN
     END
```

INORD3.S
INTIME

```
*#RUN=;HEDG/OBJ/ENTRB3.O(BCD,NOGO)
*ENTRB3.S
      SUBROUTINE ENTERB(N,IQTY,IPRI,JTIME)
C          THIS ROUTINE BACKORDERS REQUISITIONS FOR ITEM N
C            IQTY=QUANTITY PLACED ON BACKORDER
C            IPRI=1  HIGH-PRIORITY REQUISITION
C            IPRI=2  OTHERWISE   .
C            JTIME= CLOCK TIME RFQ WAS RECEIVED
       COMMON/IDBUG/ IDBUG
      COMMON/NBMAX/NBMAX
      COMMON/NLOCBK/NLOCBK
      COMMON/NBOIU/NBOIU(1)
      COMMON/NBOIR/NBOIR(1)
      COMMON/NBOTR/NBOTR(1)
      COMMON/NBOTU/NBOTU(1)
      COMMON/NBOPT/NBOPT(1)
      COMMON/IBACPT/ IBACPT(1)
     COMMON/ICANCL/ICANCL(1)
      COMMON/IDFSNB/ IDFSNB(1)
      COMMON/ ILOCBK/ ILOCBK(1)
      COMMON/ IPRIOR/ IPRIOR(1)
      COMMON/ IQTYB/IQTYB(1)
     COMMON/ITIME/ITIME
      COMMON/ITMBAC/ITMBAC(1)
C           RESERVE A STORAGE LOCATION FOR THIS INFORMATION
      IPT=ILOCBK(NLOCBK)
C
C               IF BO FILE IS FULL, CANCEL THIS REQUISITION
C                     OTHERWISE, GO TO 5 AND RECORD THIS BO.
     IF(NLOCBK.GE.1) GO TO 5
C                      THERE'S NO ROOM.  CANCEL IT.
     IF(IDBUG.GE.1) WRITE(6,7) ITIME,IQTY,IPRI
   7  FORMAT(4X,'****ENTERB--ITIME=',I8,'  ',
&      'BO FILE IS FULL.  CANCEL ',
&           ' REQ FOR ',I8,' UNITS, PRI=',I5)
C
     CALL CUM(ICANCL,IQTY,N)
     RETURN
C
   5 CONTINUE
C          UPDATE BACKORDER COUNTERS
C
      NBOTR(N)=NBOTR(N)+1
      NBOTU(N)=NBOTU(N)+IQTY
      IF(IPRI.NE.1)GO TO 10
      NBOIR(N)=NBOIR(N)+1
      NBOIU(N)=NBOIU(N)+IQTY
  10 CONTINUE
      IF(IDBUG.NE.1) GO TO 15
      WRITE(6,13)N,IQTY,IPRI,NBOIU(N),NBOTU(N),NBOIR(N),NBOTR(N),IPT
  13 FORMAT(4X,'****ENTERB--N=',I5,'  IQTY=',I5,'  IPRI=',I5,
     &'  NBOIU=',I5,'  NBOTU=',I5,'  NBOIR=',I5,'  NBOTR=',I5,
     &'IPT=',I5)
  15  CONTINUE
C        DID THIS REQUISITION CAUSE THE BACKORDER FILE TO OVERFLOW
      NLOCBK=NLOCBK-1
      IF(NLOCBK.GE.0)GO TO 20
C        WRITE ERROR MESSAGE
      WRITE(6,91)
   91  FORMAT(1H0,20X,"ERROR--BACKORDER FILE OVERFLOW,FILE DUMP ON",
     &  " NEXT PAGE")
      WRITE(6,82)
   82 FORMAT(1H1,10X,23H**BACKORDER FILE DUMP**)
      DO 83 K=1,NBMAX
   83 WRITE(6,84) K,ITMBAC(K),IDFSNB(K),IPRIOR(K),IQTYB(K),IBACPT(K)
   84 FORMAT(1H ,3X,7HREC NO=,I3,5X,7HITMBAC=,I7,5X,7HIDFSNB=,I10,5X,7HI
     &PRIOR=,I1,5X,6HIQTYB=,I7,5X,7HIBACPT=,I7)
      RETURN
C     RECORD QUANTITY,PRIORITY,FSN ID,AND TIME DATA FOR THIS BO REQ
   20 ITMBAC(IPT)=JTIME
      IDFSNB(IPT)=N
      IPRIOR(IPT)=IPRI
      IQTYB(IPT)=IQTY
C        ARE ANY OTHER BACKORDERS OUTSTANDING ON ITEM N
      IF(NBOPT(N).GT.0) GO TO 40
C        RECORD POINTER DATA
      NBOPT(N)=IPT
      IBACPT(IPT)=0
      RETURN
C        IS THE NEW BO A PRIORITY 1 REQUISITION
   40 IF(IPRI.EQ.1) GO TO 60
C        NOTE-- NEW LOW PRIORITY BACKORDERS ARE INSERTED LAST ON THE
C               BACKORDER CHAIN. THE REMAINING STEPS IN THIS PORTION
C               OF THE SUBROUTINE ACCOMPLISH THIS OBJECTIVE
C        SET JPT EQUAL TO THE FILE LOCATION NO OF THE FIRST BACKORDERED
C        REQUISITION IN THE CHAIN
      JPT=NBOPT(N)
C        IS JPT THE LAST LINK IN THE CHAIN
   49 IF(IBACPT(JPT).EQ.0) GO TO 50
      KPT=IBACPT(JPT)
      JPT=KPT
      GO TO 49
C        RECORD NEW POINTERS
   50 IBACPT(JPT)=IPT
      IBACPT(IPT)=0
      RETURN
C        SET JPT EQUAL TO LOCATION NO OF FIRST BO ON CHAIN
   60 JPT=NBOPT(N)
C        IS BACKORDER JPT A HIGH PRI BO
   61 IF(IPRIOR(JPT).NE.1) GO TO 80
C        IS JPT THE LAST BACKORDER ON THE CHAIN
      IF(IBACPT(JPT).EQ.0) GO TO 62
      KPT=JPT
      JPT=IBACPT(KPT)
      GO TO 61
C          INSERT NEW BO AS LAST LINK ON CHAIN
   62 IBACPT(JPT)=IPT
      IBACPT(IPT)=0
      RETURN
C          IS JPT THE ONLY BO ON CHAIN
   80 IF(JPT.NE.NBOPT(N)) GO TO 81
C          INSERT NEW BO AS FIRST LINK ON CHAIN
      NBOPT(N)=IPT
      IBACPT(IPT)=JPT
      RETURN
C          INSERT NEW BO AS LINK BETWEEN KPT AND JPT
   81 IBACPT(KPT)=IPT
      IBACPT(IPT)=JPT
      RETURN
      END
```

ENTRB3,S

CATALOG/FILE DESCRIPTION= HEDG/DATAB3.S

```
*#RUN=#HEDG/OBJ/DATAB3.O(BCD,8L IB)PEQS/RANDU.O,R#
*#  HEDG/OBJ/DEMPR3.O#HEDG/ID62DATA"03",R
*DATAB3.S
C
C      THIS ROUTINE SELECTS INSSIM RECORDS WITH AVERAGE ANNUAL
C      DEMANDS IN QTRS 1-8 THAT LIE IN THE RANGE RLB TO RUB.
C      REQUISITION COUNTS ARE GENERATED BY MONTE CARLO TO BE
C      CONSISTENT WITH UNIT DEMAND RECORDS.
C
C       THIS ROUTINE ASSUMES INPUT RECORDS ARE BINARY FORMAT AS
C       DEFINED IN INSSIM VOL. I, APPENDIX A.
C
      PARAMETER NOOO=33
      DIMENSION FSN(4),NOUN(2),IDEM(NOOO),IRET(NOOO),IREQ(NOOO),MGTCD(4
       CHARACTER MGTCD
       CALL FPARAM(1,132)
      READ(5,3) LOUT,LPRNT,LSKIP
    3     FORMAT(V)
      PRINT, "    LOUT =",LOUT,"  LPRNT =",LPRNT,"  LSKIP =",LSKIP
      READ(5,3)RLB,RUB
      PRINT,"      LOWER BOUND =",RLB,"     UPPER BOUND =",RUB
C
C        INITIALIZE RANDOM NUMBER SEED
C
               R=RANDU(-.1)
C
C        INITIALIZE COUNTERS
C
      KNT=0
      IOUT=0
C
      ISKIP=0
   10     CONTINUE
         ISKIP = ISKIP + 1
    15    CONTINUE
       KNT = KNT + 1
      IF(MOD(KNT,10).EQ.0) WRITE(6,3)"INPUT ITEM # ",KNT
      READ(8, END=200)ALC,FSN,IR,UM,UCOST,NOUN,MGTCD,IOH,IOR,
&         LTADM,LTPROD,IPP,RIPP
   13     FORMAT(A2,A2,A4,A6,A3,I1,A2,F9.2,A6,A4,2A1,A4,A2,
&          2I7,3I2,F4.2)
   23     FORMAT(I7,1X,2A3,A4,1X,A6,A3,1X,I2,A3,F10.2,1X,A6,A4,1X,
&          2A1,A4,A2,2I7,2X,3I3,F5.2)
   33     FORMAT(A2,A2,A4,A6,A3,I1,24I7)
   43     FORMAT(3(/10X,8I7),T95,I10,T120,F11.0)
   53     FORMAT(3(/10X,8I7),T105,I10)
   63     FORMAT(3(/10X,8I7),T115,I10)
       READ(8   )ALC,FSN,IR,IDEM
       READ(8   )ALC,FSN,IR,IRET
       READ(8   )ALC,FSN,IR,IREQ
```

```
C
C             CHECK IF THIS IS AN F104 OR F5 ITEM
C             (I.E. IS A MFQI CODE = MGTCD(3) = 403Z OR 420Z)
C              IF SO, SKIP THE ITEM.
      IF( MGTCD(3).EQ."403Z") GO TO 15
      IF( MGTCD(3).EQ."420Z") GO TO 15
C
C
C             SELECT ITEMS WITHIN THE RANGE RLB-THRU-RUB
C
      ID=0
      IR=0
      DO 80 I=1,3
      ID=ID+IDEM(I)
      IR=IR+IRET(I)
   80      CONTINUE
      TD=UCOST*FLOAT(ID-IR)/2.
C
C
      IF(TD.LT.RLB)GO TO 15
      IF(TD.GT.RUB) GO TO 15
C
      KDEM=0
      KRET=0
      KREQ=0
      DO 110 I=1,NQQQ
      KDEM=KDEM+IDEM(I)
      KRET=KRET+IRET(I)
      KREQ=KREQ+IREQ(I)
       IF(IDEM(I).GT.10000) GO TO 15
       IF(IRET(I).GT. 2000) GO TO 15
C
  110      CONTINUE
C
C               PRINT THIS ITEM
C
C
C
C           REJECT THIS ITEM IF TOTAL DEMANDS OR RETURNS EXCEED
C          20 TIMES THE UPPER-BOUND BOUND RATE.
C
               RMAX=20.*RUB
               RATE=UCOST*FLOAT(KDEM)*4./FLOAT(NQQQ)
              IF(RATE.GT.RMAX) GO TO 15
C
               RATE=UCOST*FLOAT(KRET)*4./FLOAT(NQQQ)
              IF(RATE.GT.RMAX) GO TO 15
      IF(ISKIP.LT.LSKIP) GO TO 10
      ISKIP=0
C
C                 GENERATE REQUISITION COUNTS FOR FIRST 8 QUARTERS
C
         DO 100 N=1,NQQQ
C
```

```
      IQTY=IDEM(J)
      ITOTL=0
      IRKNT=0
      IF(IQTY.LE.0)GO TO 81
C
C          GIVEN TOTAL DEMAND (IQTY) IN THE QUARTER, OBTAIN
C            THE NUMBER OF REQUISITIONS BY MONTE CARLO.
C
   60 R=RAND(.2)
C
       CALL GETREQ(R,IQTY,IRQ)
      IRKNT=IRKNT+1
      ITOTL=ITOTL +IRQ
      IF (ITOTL .LT.IQTY)GO TO 60
C
C
       IREQ(J)=IRKNT
     GO TO 100
   81 IREQ(J)=0
  100 CONTINUE
     TDOL=UCOST*FLOAT(KDEM)
     WRITE(6,23)KNT,ALC,FSN,IR,UM,UCOST,NOUN,MGTCD,
&          IOH,IOR,LTADM,LTPROD,IPP,RIPP
     WRITE(6,73)KDEM,KRET,KREQ,TDOL
   73 FORMAT(T95,3I8,F10.0)
     IF(IOUT.GT.LPRNT)GO TO 200
     WRITE(6,77)IDEM
     WRITE(6,77)IRET
     WRITE(6,77)IREQ
   77 FORMAT(10X,16I7)
C
C
C
  200     CONTINUE
C
C          OUTPUT THIS ITEM TO FILE 9
C
     WRITE(9)ALC,FSN,1,UM,UCOST,NOUN,MGTCD,IOH,IOR,LTADM,
&        LTPROD,IPP,RIPP
   83    FORMAT(I7,1X,A2,A2,A4,A6,A3,I1,A2,F9.2,A6,A4,2A1,A4,A2,
&          2I7,3I2,F4.2)
   93    FORMAT(5X,8I7)
       WRITE(9)ALC,FSN,2,IDEM
       WRITE(9)ALC,FSN,3,IRET
       WRITE(9)ALC,FSN,4,IREQ
     IOUT=IOUT+1
  300    CONTINUE
     IF(IOUT.GT.LOUT) GO TO 900
     GO TO 10
  900    CONTINUE
     WRITE(6,903)KNT,IOUT
  903    FORMAT(/////T40,'RECORDS IN  =',I5/T40,'RECORDS OUT =',I5)
     STOP
     END
```

DATAB3.S

APPENDIX B

FORTRAN SOURCE LISTINGS

```
C---BEGIN LAGRANGIAN LOOP
C
      DO 200 MRUN=1,NRUN
          IRUN=MRUN
          COSHRT=CSHORT(MRUN)
          IF(IDBUG.EQ.1)WRITE(6,8090)COSHRT
C
          8090 FORMAT(//10('****'),'BEGIN SIMULATION ',
&'WITH COSHRT =',F10.4,10('****')//)
C              REWIND ITEM INPUT FILE INLU
      REWIND INLU
C
C---RECORD RANDOM NUMBER SEED
C
C
C---INITIALIZE RANDOM NUMBER STREAM
C             INITIALIZE DEMPR3 RANDOM NUMBER STREAM
      IF(IISEED.EQ.0) RNLAST = RANDEM(-.2)
      IF(IISEED.NE.0) RNLAST = RANDEM(-ABS( FLOAT(IISEED) ) )
C
C             INITIALIZE GENERAL RANDOM NUMBER STREAM
C
C
      RNLAST = IISEED
      IF(IISEED.EQ.0) RNLAST = RANDU(-.1)
      IF(IISEED.NE.0) RNLAST = RANDU(-ABS(RNLAST))
C
C
       CALL ZERO
C
```

NEW HEDGS
MAIN
PROGRAM
CODE